Loughborough University
Institutional Repository

Adaptive transform coding for digital image communication

This item was submitted to Loughborough University’s Institutional Repository by the/an author.

Additional Information:


Metadata Record: https://dspace.lboro.ac.uk/2134/27360

Publisher: © B.M.S. Moussa

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 2.5 Generic (CC BY-NC-ND 2.5) licence. Full details of this licence are available at: http://creativecommons.org/licenses/by-nc-nd/2.5/

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
TO THE SPIRIT OF MY PARENTS

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

* وَقَضَى رَبُّكَ أَنْ تُعْبَدُوا إِلَّا إِيَّاهُ وَالْوَلِيدَانِ إِسْحَاقَ وَيَسْعَى إِلَّا بِفَطِيرٍ مُّنْفِقٍ وَأَخَافُّ لَهُمَا جَنَّاتُ النُّعُمَ وَقَلْ رَبِّ ارْضِعُهُمَا فَـلا تَقْلِ

وَعَمِّي صَفِيرًا ۱١۸۰ ذَّبَّ أَعْلَمُ أَنَّا نُفَوَّسُكَ إِنْ تَكُونُوا صَلِّيْنَ مَا كَانَ لأَوْرَابٍ غَفِّرُوا ۱٥٨
ADAPTIVE TRANSFORM CODING FOR DIGITAL IMAGE COMMUNICATION

by

B.M.S. MOUSSA, Dip., M.Sc.

A Doctoral Thesis

submitted in fulfilment of the requirements

for the award

of

the Doctor of Philosophy Degree of

Loughborough University of Technology

April, 1985

SUPERVISOR: Mr. R.J. Clarke

Department of Electronic and Electrical Engineering

© by B.M.S. Moussa, 1985
The performance of transform image coding schemes can be improved substantially by adapting to changes in image statistics. Essentially, this is accomplished through adaptation of the transform, bit allocation, and/or quantization parameters according to time-varying image statistics. Additionally adaptation can be used to achieve transmission rate reduction whilst maintaining a given picture quality.

Generally adaptive transform systems result in a variable bit rate at the output of the encoder, changing according to signal statistics. Buffer stores are therefore needed at both transmitter and receiver to allow a varying rate sequence of digits to be transmitted over a uniform rate channel, and in this thesis several approaches have been investigated in which a variable input-fixed output rate buffer in the transmitter has been simulated. Practical transmitting stores must be of finite size and therefore may be subject to overflow and underflow. To prevent these problems, a new strategy is developed. Each image frame is divided into sub-blocks which are categorized within four classes according to their activity, and the classification map is then controlled according to buffer status. Five different schemes were investigated in respect of overall best performance and buffer size. A further control strategy employing coarse quantization of the
normalized transform coefficients during periods when the buffer is close to overflow was also investigated. The best of the first, and the latter, systems are considered to be useful and simple additions to adaptive image coding schemes.

In coding of still pictures the desired average bit rate is usually achieved by iteratively adjusting the distortion parameter. In coding of moving images alternative methods can be employed, and five schemes have been investigated here to control the bit rate over the channel, all schemes being used to code an eight frame moving sequence.

For reduction of system complexity and bit rate, two techniques have been considered. The bit rate and computational burden is reduced with the first, and overhead information reduction is achieved by the second. Simulation results show acceptable picture quality.
ACKNOWLEDGEMENTS

It could almost be said that there are not enough words to express my feelings towards my colleagues, friends and relatives with regard to my work. First of all, I wish to express most deeply my sincere thanks and gratitude to my Supervisor, Mr. R.J. Clarke, for his constant guidance, help, comments, suggestions of new ideas and encouragement throughout the course of the work. Also for his patience in reading and correcting the manuscript of this work and without whom this thesis would not have been completed.

I am also grateful to the Department of Electronic and Electrical Engineering and the Computer Centre of Loughborough University of Technology for providing the facilities to carry out this work.

Special thanks also to my parents (who passed away during the course of the work) and my brothers for their continuous moral support throughout my education.

I gratefully acknowledge my sponsors, The SYRIAN ARAB people and The SYRIAN ARAB GOVERNMENT for the financial support which made this work possible.

Many thanks to my colleagues and friends at home who carried out my work there during the course of my research and who thus gave me the opportunity to carry out this work in the U.K.
Also my thanks to all my colleagues, in particular Dr. A. Najdy, Dr. H. Jormakly for important discussions on aspects of simulation at the beginning of this work, and to Mr. D. Allott for useful discussions on different aspects of image coding.

My utmost gratitude to my wife, Muna, and my children Saleem, Waseem, Dena and Amir for their understanding and patience which enabled me to complete this thesis.

I would also like to thank Mr. R.J. Clarke and Professor A.P. Clark for their help and generous hospitality throughout the duration of the course.

Finally, many thanks to the BRITISH people who cared for us during our residence in GREAT BRITAIN.
LIST OF PRINCIPAL SYMBOLS AND ABBREVIATIONS

A : The transformed image in matrix form

A_I : Energy sub-block activity index

A_{m,l} : Activity index of the absolute value of the transformed coefficients within the \((m,l)\)th sub-block location

A/D : Analogue to digital converter

a : Transformed image vector in vector space

a_i : Predictor coefficients

B : Absolute scene brightness

B_a : Eye adaptation brightness level

B_b : Level below which brightness increments are indistinguishable to the eye

B_{TX} : Transmitter buffer size

B_{RX} : Receiver buffer size

br : Transmission bit rate per element

C : Normalization factor

CCR : Compression ratio in transform domain

CM(\cdot) : Classification map for a particular frame
CON : Normalization coefficient
D : Distortion parameter
D₁ : Initialised distortion parameter
DCT : The discrete cosine transform
DR₉ : Time delay in receiver buffer to prevent distortion within delay system
DS₉ : Overall time delay introduced within delay operated system
DS₁ : Overall time delay introduced within pattern-insertion system
DS₁ₛ : Time delay smaller than DS₁
DT₉ : Time delay in the transmitter buffer to prevent underflow
DT₉ₛ : Time delay smaller than DT₉
dᵢ : Quantizer decision levels
dᵢ(\cdot) : Number of digits inserted in transmitter buffer
Eₘ₁ : The a.c. energy within the (m,l)th sub-block
eₙ : Difference signal between input signal and predicted signal in DPCM system
e'ₙ : Quantized difference signal in DPCM
E [ ] : Expected value of variables in [ ]
F : The image array in matrix form
f : Image vector in vector space
f(.) : Two-dimensional light intensity function of an image
G : Forward transform matrix
G^T : Matrix transpose of G
G^* : Complex conjugate matrix of G
g_1, g_2, g_3 : Separable transform kernels
H : Inverse transform matrix
h_1, h_2, h_3 : Inverse kernels
IBI : Binary codeword length
IB : Number of bits for a particular coefficient
IDCT : Inverse discrete cosine transform
INPADD : Transmitter buffer input address
IOUTADD : Transmitter buffer output address
INRX : Receiver buffer input address
IOUTRX : Receiver buffer output address
IQ : Number of quantization levels (in flow charts)
i(.) : Illumination level incident on the scene being viewed
L : Number of quantization levels
$L_{\text{min}}$ : Minimum value of grey level intensity
$L_{\text{max}}$ : Maximum value of grey level intensity
l : Grey level of the image at a particular point
$ar{L}$ : Average codeword length
MAR : Memory address register
MDR : Memory data register
$N_{bi}$ : Number of bits to be transmitted per block of class i
n : Number of elements per block
$N_{b.k}(.)$ : Number of bits assigned to the (.)th location coefficient with class K
NMSE : Normalized mean square error
pdf : Probability density function
p(.) : Probability density function of (.)
Q : Quantizer
R : Fixed transmission rate (bits/second)
\( R(t) \) : Instantaneous output of the adaptive variable bit rate encoder (bits/unit time)

\( R_{\text{cum}}(t) \) : Total number of information samples stored in the transmitter buffer up to time \( t \)

\( R(D) \) : Rate distortion function.

\( r(.) \) : The amount of light reflected by objects within a scene

\( \text{SNR} \) : Signal to noise ratio

\( \text{SQNR} \) : Signal to quantization noise ratio

\( T_i \) : Time interval between the instants at which the encoder generates the \( i \)th and \((i + 1)\)th data digits

\( T(.) \) : Transform coefficients in one, two or three dimensions

\( \hat{T}(.) \) : Inverse normalization coefficients

\( \text{Th}_1, \text{Th}_2 \) : Threshold values to prevent buffer overflow

\( \text{Th}_3 \) : Threshold value to prevent buffer underflow

\( \text{Th}_c \) : Threshold value for classification map comparison

\( \text{Th}_R \) : Threshold delay in the receiver buffer

\( \tau \) : Transmission time period \( \frac{1}{R} \)

\( W \) : Spatial light intensity distribution (cycle/degree)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WF</td>
<td>Weighting factor</td>
</tr>
<tr>
<td>X</td>
<td>Quantizer input</td>
</tr>
<tr>
<td>$X'_n$</td>
<td>Output signal of DPCM system</td>
</tr>
<tr>
<td>Y</td>
<td>Quantizer output</td>
</tr>
<tr>
<td>$Y_i$</td>
<td>Quantizer reconstruction levels</td>
</tr>
<tr>
<td>$\sigma^2_e$</td>
<td>Variance of prediction error ($l_n$)</td>
</tr>
<tr>
<td>$\sigma^2_{m,l}$</td>
<td>Variance of the $(m,l)$th sub-block</td>
</tr>
<tr>
<td>$\sigma^2_{u,v}$</td>
<td>Variance of transform coefficient in $(u,v)$ dimensions</td>
</tr>
<tr>
<td>$\hat{\sigma}^2$</td>
<td>Estimated variance of transform coefficients</td>
</tr>
<tr>
<td>$\sigma'_K(.)$</td>
<td>Normalization coefficients of the location $(.)$ for class $K$</td>
</tr>
<tr>
<td>$\mu_{m,l}$</td>
<td>Mean value of transform ac coefficients for the $(m,l)$th sub-block location</td>
</tr>
<tr>
<td>$\mu_{u,v}$</td>
<td>Mean value of transform coefficient of the $(u,v)$th location</td>
</tr>
<tr>
<td>$\delta_i$</td>
<td>Increase or decrease in buffer content due to a block of class $i$</td>
</tr>
</tbody>
</table>
# CONTENTS

## SYNOPSIS

## ACKNOWLEDGEMENTS

## LIST OF PRINCIPAL SYMBOLS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>CHAPTER I - GENERAL INTRODUCTION AND ORGANISATION OF THE THESIS</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>1.2 ORGANISATION OF THE THESIS</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER II - INTRODUCTION TO THE CODING OF VIDEO SIGNALS</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 INTRODUCTION</td>
<td>10</td>
</tr>
<tr>
<td>2.2 CLASSIFICATION OF CODING TECHNIQUES</td>
<td>16</td>
</tr>
<tr>
<td>2.2.1 Pulse Code Modulation (PCM)</td>
<td>16</td>
</tr>
<tr>
<td>2.2.2 Predictive Coding</td>
<td>18</td>
</tr>
<tr>
<td>2.2.3 Transform Coding</td>
<td>22</td>
</tr>
<tr>
<td>2.2.4 Hybrid Coding</td>
<td>26</td>
</tr>
<tr>
<td>2.2.5 Interpolative Coding</td>
<td>28</td>
</tr>
<tr>
<td>2.2.6 Contour Coding</td>
<td>29</td>
</tr>
<tr>
<td>2.3 AN IMAGE MODEL</td>
<td>30</td>
</tr>
<tr>
<td>2.4 PSYCHOVISUAL PROPERTIES OF VISION</td>
<td>32</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.4.1</td>
<td>The Human Eye</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Brightness Adaptation and Discrimination</td>
</tr>
<tr>
<td>2.5</td>
<td>MEASURES OF PICTURE QUALITY</td>
</tr>
<tr>
<td>2.5.1.</td>
<td>Image Fidelity Measures</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Subjective Rating-scale Method for Image Quality</td>
</tr>
<tr>
<td>2.6</td>
<td>COMMENTS AND CONCLUSIONS</td>
</tr>
<tr>
<td></td>
<td><strong>CHAPTER III - THE THEORY OF ADAPTIVE TRANSFORM IMAGE CODING</strong></td>
</tr>
<tr>
<td>3.1</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>3.2</td>
<td>IMAGE TRANSFORMS</td>
</tr>
<tr>
<td>3.2.1</td>
<td>General Formulation and Representation</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Types of Image Transforms</td>
</tr>
<tr>
<td>3.2.2.1</td>
<td>Fourier Transform</td>
</tr>
<tr>
<td>3.2.2.2</td>
<td>The Hadamard Transform</td>
</tr>
<tr>
<td>3.2.2.3</td>
<td>The Karhunen-Loeve (Hotelling) Transform</td>
</tr>
<tr>
<td>3.2.2.4</td>
<td>The Haar Transform</td>
</tr>
<tr>
<td>3.2.2.5</td>
<td>The Slant Transform</td>
</tr>
<tr>
<td>3.2.2.6</td>
<td>The Discrete Cosine Transform (DCT)</td>
</tr>
<tr>
<td>3.3</td>
<td>OPTIMISATION OF PARAMETERS</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Selection of Transform</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Selection of block size</td>
</tr>
</tbody>
</table>
CHAPTER IV - ADAPTIVE TRANSFORM CODING TECHNIQUES

4.1 INTRODUCTION

4.2 ADAPTIVE COEFFICIENT SELECTION AND QUANTIZATION - A REVIEW

4.2.1 Threshold Sampling

4.2.2 Zonal Sampling

4.2.3 Zonal Coding

4.2.4 Other Adaptive Transform Coding Schemes

4.2.5 Adaptive Transform Coding Method Using An Activity Index

4.3 ADAPTIVE TRANSFORM CODING USING THE DCT AND ACTIVITY INDEX CLASSIFICATION VIA A SIMPLIFIED ENERGY ESTIMATION METHOD

4.3.1 Introduction

4.3.2 The Cosine Transform

4.3.3 Statistical Properties of Cosine Transform Samples

4.3.3.1 Statistical Mean and Variance

4.3.3.2 Probability Density Model

4.3.4 Quantization

4.3.4.1 The Optimum Mean Square or Lloyd-Max Quantizer

4.3.4.2 The Compander

4.3.5 Adaptive Cosine Transform Coding Schemes For Monochrome Images
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.5.1</td>
<td>System Description</td>
<td>114</td>
</tr>
<tr>
<td>4.3.5.2</td>
<td>The Classifier</td>
<td>117</td>
</tr>
<tr>
<td>4.3.5.2.1</td>
<td>Classification Based On a.c. Energy</td>
<td>117</td>
</tr>
<tr>
<td>4.3.5.2.2</td>
<td>Classification Based On The Weighted Sum Of The Absolute Values Of The Transform Coefficients</td>
<td>128</td>
</tr>
<tr>
<td>4.3.5.2.3</td>
<td>Bit Assignment</td>
<td>135</td>
</tr>
<tr>
<td>4.3.5.3</td>
<td>Overhead Information</td>
<td>143</td>
</tr>
<tr>
<td>4.4</td>
<td>EXPERIMENTAL RESULTS</td>
<td>144</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Threshold Sampling</td>
<td>145</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Zonal Sampling</td>
<td>150</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Activity Index Classification Via A Simplified Energy Estimation Method</td>
<td>150</td>
</tr>
<tr>
<td>4.5</td>
<td>DISCUSSION</td>
<td>169</td>
</tr>
<tr>
<td>4.6</td>
<td>CONCLUSION</td>
<td>170</td>
</tr>
</tbody>
</table>

CHAPTER V - BUFFERS FOR VARIABLE BIT RATE SYSTEMS 172

5.1  INTRODUCTION 173
5.2  TYPES OF BUFFER STORAGE 182
5.3  BUFFERING IN ADAPTIVE IMAGE PROCESSING 185

5.3.1  THE BUFFER UNDERFLOW PROBLEM 185

5.3.1.1  Delay Within The Transmitter Buffer To Prevent Underflow 187
5.3.1.2  Pattern Insertion To Prevent Underflow Within The Transmitter Buffer 190
5.3.1.3  Delay In The Receiver Buffer To Prevent Distortion Within The Delay System 192
5.3.1.4  Overall System Delay For the Pattern-Insertion System 197
| 5.3.1.5 | Comparison Of The Overall Delays Involved In The Two Systems | 200 |
| 5.3.2 | BUFFER SIZE ESTIMATION AND THE OVERFLOW PROBLEM | 203 |
| 5.3.3 | EXPERIMENTAL PROCEDURES | 208 |
| 5.3.3.1 | The Adaptive Transform Coding System And Buffer Simulation | 210 |
| 5.3.3.1.1 | The Transmitter Buffer Input | 214 |
| 5.3.3.1.2 | The Transmitter Buffer Output With And Without Delay | 214 |
| 5.3.3.1.3 | The Transmitter Buffer Output With Pattern-insertion | 217 |
| 5.3.3.1.4 | The Receiver Buffer Input | 220 |
| 5.3.3.1.5 | The Transmitter Buffer Input With Classification Map Control According To Buffer Status | 220 |
| 5.3.3.1.6 | Receiver Buffer Output | 228 |
| 5.3.3.1.7 | The Buffer Structure | 231 |
| a) Transmitter Buffer | | |
| b) Receiver Buffer | | |
| 5.3.4 | EXPERIMENTAL RESULTS | 235 |
| 5.3.4.1 | Buffering With Delay (Scheme I) | 237 |
| 5.3.4.2 | Buffering With Pattern-insertion (Scheme II) | 243 |
| 5.3.4.3 | Buffering With Pattern-insertion And Classification Map Control According To Buffer Status (Schemes IIIa and IIIb) | 247 |
| 5.3.4.4 | Buffering With Delay, Insertion and Classification Map Control According To Buffer Status (Schemes IVa and IVb) | 252 |
| 5.3.4.5 | Buffering With Coarse Quantization | 263 |
CHAPTER VII - INTRA-FRAME AND INTER-FRAME ENERGY ESTIMATION

7.1 INTRODUCTION
7.2 INTERFRAME ENERGY ESTIMATION USING THE CLASSIFICATION MAP (SCHEME I)
7.3 INTRAFRAME ENERGY ESTIMATION AT THE RECEIVER (SCHEME II)
7.4 DISCUSSIONS AND CONCLUSIONS

CHAPTER VIII - RECAPITULATION AND SUGGESTIONS FOR FURTHER WORK

8.1 INTRODUCTION
8.2 THRESHOLD SAMPLING
8.3 ZONAL SAMPLING
8.4 ACTIVITY INDEX CLASSIFICATION VIA A SIMPLIFIED ENERGY ESTIMATION METHOD
8.5 BUFFERING FOR VARIABLE BIT RATE SYSTEMS
8.6 ADAPTIVE TRANSFORM CODING OF MOVING IMAGES WITH VARIABLE BIT RATE BUFFERING
8.7 INTRA-FRAME AND INTER-FRAME ENERGY ESTIMATION
8.8 SUGGESTIONS FOR FURTHER WORK

APPENDIXES:

APPENDIX I Some Properties of Matrices
APPENDIX II Simulator For An Adaptive Transform Image Coding-Decoding System With Variable Rate Buffer For Still Images (Chapter IV and V)
APPENDIX III  Simulator For An Adaptive Transform Image Coding-Decoding System With Variable Rate Buffer For Moving Image Sequences (Chapter VI)

REFERENCES
CHAPTER I

GENERAL INTRODUCTION
AND
ORGANIZATION OF THE THESIS
1.1 INTRODUCTION

God's graces are uncounted in man and perhaps the most important and valuable attributes that he possesses are vision and hearing. After Mohammed's message in 611 (the last prophet), the Arabs discovered some of the features of the human eye which were then transferred over all the world. During man's technological development he discovered the phenomenon of electric current, which led to the invention of the telephone by A.G. Bell in 1876, the first electrical system for transmission of speech signals over long distances. In 1883 the father of television P. Nipkow showed mankind the possibilities of his invention. In recent years image data transmission techniques have attracted many researchers and have received attention in different fields, and the trend with video processing nowadays is, of course, to digital storage and/or transmission rather than analogue, because of a number of advantages which it possesses as compared with the latter form.

The transmission of image information by digital communication links requires a digital data stream. Therefore the output of continuous source must be reformed into a discrete set of variables that can be coded for transmission over the channel. However, the conversion process from analogue to digital form results in an increased bandwidth requirement for transmission, in addition to a large memory storage requirement. For example, a 5.5 MHz television signal sampled
at the Nyquist rate with 8 bits per element would require a bandwidth of about 45 MHz when transmitted in digital form, and for an image of 512 x 512 picture elements at 8 bits/element intensity resolution and 25 frames per second, a rate of nearly 52 M bits/second is required.

The main problem in the design of image coding systems for digital communication links is therefore the choice of suitable coding method (or compression algorithm) which will minimize the number of code symbols required to represent image data without significant degradation in the quality of their decoded versions. Such efficient coding systems must also be implementable at a reasonable cost. Many different techniques are available for this purpose such as pulse code modulation (PCM), differential pulse code modulation (DPCM), transform coding, hybrid coding, and interpolative and contour coding. Transform image coding was invented in the late 1960's and since that time much work has been carried out in this field [1-6]. The initial concepts were developed using the Fourier transform [7-11], where the coefficients of a two-dimensional Fourier transform are transmitted over the channel. Subsequently the Hadamard transform was used [12-13], and also the Haar transform [14-16]. The discrete Karhunen-Loeve transform [17-19], which is an optimum transform, has also been used. Later, Ahmed [41] showed that the discrete cosine transform approaches the efficiency of the K-L transform for Markov process image data and Clarke [3] has shown that the DCT is the KLT where
the correlation between picture elements $p+1$. Jain [28] has suggested a sine transform. However, it has been shown [30] that the discrete sine transform (DST) has not a constant zero-order basis vector and some of the higher order basis vectors do not have zero mean values. This causes an undesirable distribution of data mean energy within what should be purely a.c transform coefficients, in contrast to the situation with the DCT, for example, where DC and a.c components are properly separated. This leads to a severe degradation in data compression performance, since artificially large a.c components are generated which affect the optimum bit allocation and quantization procedures.

The aim of the transform algorithm is to concentrate most of the energy in the original image into a few low order a.c coefficients, and this concentration of energy will occur if the image is not highly detailed. Usually the image is divided into subpictures (sub-blocks), and each individual sub-block is transformed into a set of weakly correlated coefficients (it is only possible to generate a set of almost uncorrelated transform coefficients, because it is difficult to obtain very detailed image statistics). The coefficients are then quantized and coded digitally for transmission over the channel. In general, the performance of any kind of transform system depends on a number of parameters, such as the ability to achieve the least mean-square error with acceptable subjective quality, the quantization strategy employed in a particular system.
(the efficiency with which numbers of bits are used to code the coefficients), the block size, where the mean-square performance should improve with its increase since the number of correlation terms taken into account increases, and the shape of the blocks (transforming two-dimensional arrays yields better performance than using one-dimension only, and the transform operation can also be extended to the processing of three-dimensional data moving images).

The efficiency of transform coding depends on the transform used. For example, the K-L transform is regarded as optimal, giving least mean-square error, i.e. minimum distortion in the reconstructed image, but is difficult to implement. On the other hand, the discrete cosine transform (DCT) is a suboptimum transform in the sense that it produces correlated coefficients for highly detailed images, but it is easier to implement. In addition, it has been shown that transform coding achieves a higher degree of data compression than predictive coding for a one-dimensional Markov process [93]. On real images, the two schemes are found to perform quite closely at very low distortion, but transform coding is distinctly better at high values of distortion. In practice, only adaptive predictive coding algorithms can achieve the efficiency of even non-adaptive transform coding methods [93].

In general, using the discrete cosine transform (DCT) and adaptive techniques, efficient image coding can readily be achieved. However, adaptive transform image coding systems necessitate a variable bit rate stream at the output of the
encoder. For a fixed rate transmission channel, buffers are needed at the transmitter and receiver. Practical buffers should be finite in size, and therefore they have underflow and overflow problems. These problems are very important in the context of such systems and unfortunately have received little attention in the past.

In this thesis the object of the investigation is to produce an adaptive transform coding system that will allow the use of as low a bit rate as possible without significant degradation of the reconstructed picture. Since adaptive coding systems are, in general, more complex than non-adaptive ones, possible simplifications of adaptive techniques using activity indices have been examined during the research programme. More specifically, a new classifier has been developed for such systems. Furthermore buffering problems have been studied and the simulation of transmitter and receiver buffers have been carried out, resulting in a new strategy for overcoming underflow and overflow problems by the means of classification map control according to buffer status for both still and moving images. Furthermore, activity index systems require overhead information to be transmitted over the channel to the receiver. This problem has also been examined in connection with intraframe and interframe adaptive transform coding techniques.
1.2 ORGANISATION OF THE THESIS

Chapter II contains a brief discussion of coding of video signals and the advantages of digital transmission over the corresponding analogue form.

Chapter III provides a summary of the theory of adaptive transform image coding and of the various image transforms and their properties in order to establish a basis for the coding technique. Having selected a suitable transform (the Discrete Cosine Transform) Chapter IV deals with adaptive transform coding schemes proper, and starts by introducing the adaptivity considerations that need to be taken into account. A review of adaptive transform coding techniques that have recently been used is presented and, following this, the simulation of two-dimensional adaptive transform coding using activity index classification via a simplified method is described and the results discussed.

Although having achieved a substantial reduction in activity index computation, activity index systems still produce a variable bit rate at the output of the encoder. Chapter V describes buffer simulations for variable bit rate systems. Various buffering schemes are investigated and simulated and the results obtained using different source pictures are discussed. Underflow and overflow problems are considered and a new technique is developed to prevent buffer underflow and overflow with a reasonable buffer size.
In Chapters IV and V bit rate reduction is achieved for still images where the bit rate is controlled iteratively. In Chapter VI, alternative methods for bit rate control with moving image sequences are examined. Different schemes are investigated, taking into account interframe activity, and buffering techniques are incorporated into systems covered in this chapter.

Chapter VII deals with the inter and intra-frame energy estimation. The advantages of activity index systems are employed to reduce the number of computational operations and transmission bit rate. In addition intra-frame energy estimation at the receiver has been investigated to reduce the amount of overhead information and the buffer size.

Chapter VIII concludes this thesis, and summarises the ideas, and work that has been carried out during the course, of the research programme. The thesis ends with suggestions for further work.
CHAPTER II

INTRODUCTION TO THE CODING OF VIDEO SIGNALS
2.1 INTRODUCTION

For the transmission of information from a continuous source over a digital channel, the output from the source must be represented by a set of discrete variables that can subsequently be coded before being applied to the channel input. The usual procedure is to sample the continuous source output at regular time intervals, quantize the amplitude of the samples, and then apply a suitable coding technique to the discrete time, discrete amplitude samples. The digitized information has advantages over the corresponding analogue form in terms of:-

- Processing flexibility
- High noise immunity
- The possibility of encryption.

And therefore all modern coding methods are orientated towards digital transmission.

There is nowadays a large amount of point-to-point digital transmission of information, and of particular interest in the present context is the transmission of picture material.

For example:
- International satellite links transmitting live programmes around the world
video-conferencing services
- Facsimile transmission of newspapers and printed material
- Satellites beaming a continuous stream of meteorological photographs and earth-resource pictures to the ground.

There are also a number of important military applications such as the control of remotely piloted vehicles, operating in a jamming environment, and the transmission of aircraft, radar, sonar, and computer data.

Figure 2.1 shows a general block diagram of an image transmission system, which contains an image source consisting of a television camera or facsimile scanner, followed by a source coder that transforms the source data into a form with minimal transmission requirements. The output from the source coder is then converted to a format suitable for transmission. This step involves the modulation of the transmitted carrier and, often, error protection coding against channel errors caused by noise. The channel decoder and source decoders carry out the inverse of the coding process to produce a reconstructed image for display at the image destination.

All realistic channels, over which information is to be transmitted, have a limited capacity, because of restricted bandwidth and the presence of noise. An efficient coder
FIG. 2.1 GENERAL BLOCK DIAGRAM OF IMAGE TRANSMISSION SYSTEM.
is one which enables the transmission of as much information as possible over such channels. Therefore the aim of one approach to coding is to reduce the required transmission rate for a given picture quality without unduly complicating the coding system. This is known as "low bit rate coding". Efficient coding is usually achieved in three stages as shown in Figure 2.2.

Stage One (Representation)

Here the original signal is represented in an appropriately altered form which might, for example, in the case of transform coding, be a set of transform coefficients. This operation is generally reversible. Inherent statistical redundancy is also reduced.

Stage Two (Quantization)

The accuracy of representation of the data is reduced in this stage while still meeting the required picture quality objectives\[20\]. For example, darker portions of a picture may be coded more accurately than bright portions, to acknowledge the fact that the visual system is more sensitive to small intensity changes in darker areas. This operation is irreversible.

Stage Three (Word Assignment)

In this stage statistical redundancy in the signal can be further reduced by exploitation of the remaining correlation
between error elements, or transform coefficients.

A Huffman code[45] may be used to assign short code words to values that occur more frequently and longer code words to values that occur rarely. In this case the operation is reversible.

When the aim of coding is to represent the signal more efficiently, the effect of errors in the transmission channel becomes more serious. Consequently it is frequently necessary to reintroduce a controlled form of redundancy into the signal in the channel encoding process in order to reduce the impact of such errors. Figure 2.3 shows the overall system divided into source encoding, in which redundancy is removed from the signal for the purpose of achieving more efficient data representation, and channel coding where redundancy is reinserted into the signal in order to obtain better channel-error performance.

In practice, the overall relation between efficient picture coding systems and practical transmission channels is one of economic trade-off in design, - balancing picture quality, circuit complexity, bit rate, and error performance.
FIG. 2.2 BLOCK DIAGRAM OF ENCODING PROCESS.

FIG. 2.3 SOURCE-CHANNEL ENCODING.
2.2 CLASSIFICATION OF CODING TECHNIQUES

Coding techniques that have been used for picture coding can be classified as shown in Table 2.I\textsuperscript{[21]}.

2.2.1 Pulse Code Modulation (PCM)

PCM coding is the simplest, most basic form of image coding. In this system the image signal is sampled generally at the Nyquist rate (time discreteness), then quantized using a sufficient number of quantization levels (amplitude discreteness), and coded in binary form for transmission. For binary facsimile transmission, the image signal is quantized to only two levels, black or white, and coded with 1 bit per sample. Monochrome imagery is usually quantized with from 64 to 256 levels per sample (corresponding to 6 or 8 bits) and uses fixed-length, binary code words. The number of quantization levels for monochrome images in PCM systems must be large enough so that the quantization noise will not cause a grey scale contouring effect in the reconstructed image. The mean square quantization noise for a uniform quantizer with step size $\delta$ is given by

$$E_q^2 = \frac{\delta^2}{12}$$  \hspace{1cm} (2.1)

For a sinusoid $E_s \sin \omega_0 t$ having a mean square value of $E_s^2$, then provided that the number of quantization levels $2^N$
TABLE 2.1 PICTURE CODING METHODS.
is large and no peak overload occurs, the peak signal to quantization noise ratio (SQNR) can be approximately represented by

\[ \text{SQNR} = 6.02n + 1.8 \text{ dB} \] (2.2)

where \( n \) is the number of bits used to code a word. A value from 38 dB to 50 dB is required to obtain satisfactory subjective quality when \( n \) is between 6 and 8 bits per picture element. Figure 2.4 shows a simple diagram of the PCM encoding system. PCM coding is inefficient a) because it ignores the spatial dependency among elements and b) because it treats all quantized amplitude levels as equally likely.

2.2.2 Predictive Coding

The general block diagram of a predictive image coding system is shown in Figure 2.5. Predictive coding is also known as DPCM (Differential Pulse Code Modulation). In such a system, the amplitude of each element is predicted on the basis of the history of previously scanned elements. Then the predicted estimate \( \hat{X}_n \) is subtracted from the actual element amplitude \( X_n \) and the difference signal \( e_n \) is quantized, coded and transmitted. The input signal \( X_n \) is an 8-bit PCM word. For the \( n \)th input sample \( X_n \), the linear predictor generates a prediction value \( \hat{X}_n \) which is calculated from \( k \) previous samples according to the relation
ANALOGUE AMPLITUDE LEVELS

- SAMPLING AT NYQUIST RATE
- QUANTIZATION
- BINARY WORD ASSIGNMENT

BAND LIMITED VIDEO INPUT

PCM CODED SIGNAL

FIG. 2.4 PCM ENCODING. (a) BLOCK DIAGRAM OF A PCM ENCODER (b) FOUR-BIT BINARY REPRESENTATION OF AMPLITUDE LEVELS BETWEEN 0 AND 15.
\[
\hat{X}_n = \sum_{i=1}^{k} a_i X_{n-i} \quad (2.3)
\]

where \( i = 1, 2, \ldots, k \), \( k \) is the order of predictor, and \( a_i \) are the predictor coefficients. The predictor coefficients are optimised to yield a prediction error

\[
e_n = X_n - \hat{X}_n \quad (2.4a)
\]

or

\[
e_n = X_n - \sum_{i=1}^{k} a_i X_{n-i} \quad (2.4b)
\]

with the minimum variance, which given by

\[
\sigma_e^2 = \mathbb{E}[X_n - \sum_{i=1}^{k} a_i X_{n-i}]^2 \quad (2.4c)
\]

The prediction error \( e_n \) is then quantized to produce \( e'_n \) and coded for transmission. At the receiver the quantized difference signal \( e'_n \) is used to form a reconstruction \( \hat{X}_n \) of the original image by summing the receiver prediction \( \hat{X}_n \) and the quantized difference signal \( e'_n \). The output signal \( X'_n \) differs from the input signal \( X_n \) by the quantization error, \( q_n \), i.e.

\[
X'_n = X_n + q_n \quad (2.5)
\]

Predictive coding techniques make use of the picture signal statistics to reduce the bit rate. Television signals are highly correlated, both spatially and temporally.
FIG. 2.5 BLOCK DIAGRAM OF A DPCM SYSTEM.
Correlation, or linear statistical dependence, indicates that a linear prediction of sample values based on the values of the neighbouring elements, will result in prediction errors that have a smaller variance than the original element. Owing to the smaller variance of signal to be quantized, coded and transmitted, the amplitude range of the quantizer and thus the number of quantization levels can be reduced, resulting in a requirement for fewer bits per element than for a PCM system of the same SNR.

The simplest form of predictive image coder is the delta modulation system [93] in which the prediction is formed from the previous prediction value quantized to only two levels. More usual differential pulse-code-modulation systems utilise previous element prediction with from 4 to 16 quantization levels allocated to each prediction difference sample.

2.2.3 Transform Coding

In transform coding systems, such as that shown in Figure 2.6, a unitary transform is taken over an entire image, or repeatedly over subsections called blocks. For this purpose, Fourier, sine, cosine, Hadamard, Haar, Slant, Karhunen-Loeve transforms, and later high correlation and low correlation transforms have been extensively utilised [31],[76].

Correlation between the elements suggests that bit rate reduction can be achieved by transforming the spatial
FIG. 2.6 BLOCK DIAGRAM OF A TRANSFORM CODING SYSTEM.
data into a set of independent coefficients, which can then be individually quantized according to some fidelity criterion. The transformations used are linear and unitary. For the purpose of transformation, the image is divided into subpictures, and this process means that the redundancy existing between the subpictures is neglected. Therefore, purely on a statistical basis, it is advantageous to have large subpictures. However, for implementational simplicity, as well as to be able to exploit local changes in picture statistics and visual fidelity, a smaller subpicture is desirable. Also, since most of the compression results from dropping coefficients with small energy, an important consideration for an efficient transform is the amount of energy compaction achievable. Data compression in transform coding techniques can be achieved by selecting those transformed samples for transmission in accordance with a zonal or threshold sampling strategy. In threshold sampling, a coefficient is transmitted if its magnitude is greater than a predetermined threshold value.

For zonal sampling, only these coefficients lying in a prespecified zone or zones (usually covering low-spatial frequencies), are selected for transmission. Each coefficient zone in each transform subpicture is then quantized and coded. Transform samples in each zone are quantized with the same number of quantization levels, which is set proportional to the expected variance of transform coefficients.
The zonal coding scheme might also be used in association with a block or zone classification technique. In such a technique the transform blocks or zonal elements are sorted into classes by the level of image activity present. Within each activity (ac energy) level, coding bits can be allocated to individual transform elements within a block, zone or zones according to the variance matrix of the transformed data. Bits are then adaptively distributed between more and less highly detailed image areas to provide the desired efficiency of a zonal adaptive coding technique.

For a constant-word length code, the number of bits assigned to each coefficient is set proportional to the logarithm of the variance of the coefficient, which may be determined for a given spatial domain correlation model. Probability density models of the transform coefficients are utilised to select quantization levels to minimise the mean-square coding error.

At the receiver the incoming bits are decoded, and the quantized transform coefficient array is reconstructed. Subsequently an inverse transformation produces the reconstructed image array. With these techniques monochrome images may be coded at rates of 1.0 to 1.5 bits/element with mean-square errors of less than 0.5%. Colour images require an additional 0.5 to 1.0 bit/element.
2.2.4 Hybrid Coding

In transform coding techniques, it is assumed that successive blocks of data are independent. Indeed, if the block size is large, for example, 32 or larger, the average interblock correlation is negligible. However, the use of a large block size, although it can achieve a greater reduction in bit rate, suffers from two distinct disadvantages:

1. It requires more computation time, more complex implementation and storage of large amounts of data both at the transmitter and the receiver, and consequently produces a delay in transmission.

2. Image statistics may vary widely within a block if the block size is large. Adaptive coding to match statistics within a block is then difficult to accomplish.

These drawbacks can be mitigated by choosing a small block size and then applying coding schemes which exploit interblock redundancy. One natural way is to apply predictive coding to exploit the redundancy between the transform coefficients of different blocks. This type of scheme, comprised of transform coding and predictive coding, is called hybrid coding. Figure 2.7 shows a block diagram of hybrid coder and decoder. Specifically, three types of schemes have been considered:

1. A one-dimensional block along a horizontal line with
FIG. 2.7 BLOCK DIAGRAM OF A HYBRID TRANSFORM /DPCM ENCODER
DPCM in the vertical direction.

2. A small two-dimensional block, with DPCM using coefficients of the previous horizontal block for prediction.

3. For interframe coding, a two-dimensional block, and DPCM in the temporal direction.

The saving in storage and computation when using hybrid coding is considerable compared with transform coding.

2.2.5 Interpolative Coding

In this technique, a subset of picture elements is transmitted and the remaining elements are interpolated. As an example, Figure 2.8 shows a case in which there is a 2 : 1 subsampling of picture elements along each scan line.

![Interpolative Coding Diagram]

**Fig. 2.8** - An example of interpolative coding using 2:1 horizontal subsampling, which is staggered from one line to the next.
The sub-sampled elements are staggered from one line to the next and are interpolated by a four-way average as shown by arrows (e.g. element A is interpolated by averaging elements B, C, D and E).

Most interpolation methods have used weighted averages using either straight lines or higher degree polynomials [22-24]. It appears that interpolation using straight lines is quite effective and not much is gained by interpolation using polynomials of higher degree. Switched interpolation may be more effective than the fixed interpolation. As an example, in Figure 2.8, pel A may be reconstructed by the following rule:

\[
\hat{A} = \begin{cases} 
0.5 \ (C + D), & \text{If } |C - D| \leq |B - E| \\
0.5 \ (B + E), & \text{otherwise}
\end{cases}
\]

2.2.6 Contour Coding

In contour coding [21],[25],[46], a picture is separated into two parts: edges and remaining 'texture'. Such a separation is hypothesized to do well, since by isolating edges and reproducing them accurately, the picture looks sharp and detailed. Spatial continuity of edge points (sensitivity of eye to edges) permits the use of algorithms developed for two-level (facsimile) data.

Early schemes separated the picture into "highs" and "lows", and used a contour-tracing type algorithm for coding the "highs". The "lows" part of the signal was then subsampled and coded with higher amplitude accuracy.
2.3 AN IMAGE MODEL

The term image refers to a two-dimensional light intensity function, which might be denoted by \( f(x,y) \), where the value or amplitude of \( f \) at spatial coordinates \((x,y)\) gives the intensity (brightness) of the image at that point. Since light is a form of energy, \( f(x,y) \) must be non-negative and finite, that is

\[ 0 \leq f(x,y) < \infty \quad (2.6) \]

The images we perceive in our everyday visual activities normally consist of light reflected from objects. The basic nature of \( f(x,y) \) may be considered as being characterized by two components. One component is the amount of source light incident on the scene being viewed, while the other is the amount of light reflected by the objects in the scene. These components are appropriately called the illumination and reflectance components, and are denoted by \( i(x,y) \) and \( r(x,y) \) respectively. These functions are combined as product for form \( f(x,y) \):

\[ f(x,y) = i(x,y) \cdot r(x,y) \quad (2.7) \]

where \( 0 \leq i(x,y) < \infty \) \( (2.8) \)

and \( 0 \leq r(x,y) < 1 \) \( (2.9) \)

Sometimes the intensity of a monochrome image \( f \) at coordinates \((x,y)\) is called the grey level \( l \) of the image at that point, and \( l \) lies in the range...
\( L_{\text{min}} \leq l \leq L_{\text{max}} \) \hspace{1cm} (2.10)

In theory, the only requirement in \( L_{\text{min}} \) is that it be nonnegative, and \( L_{\text{max}} \) that it be finite. In practice \( L_{\text{min}} = i_{\text{min}} r_{\text{min}} \) and \( L_{\text{max}} = i_{\text{max}} r_{\text{max}} \). The interval \([L_{\text{min}}, L_{\text{max}}]\) is called the grey scale. It is common practice to shift this interval numerically to the interval \([0, L]\), where \( l = 0 \) is considered black and \( l = L \) is considered white in the scale. All intermediate values are shades of grey varying continuously from black to white.
2.4 PSYCHOVISUAL PROPERTIES OF VISION

The human eye is the most considerable part of an image processing system. Because it is the end user of such a system, an understanding of the eye mechanism is very useful for the system designer. Such knowledge can be utilized to develop conceptual models of the human visual process, and these models are useful for the design of image processing systems as well as in the setting up of measures of image fidelity and intelligibility.

2.4.1 The Human Eye

A horizontal cross section of the human eye is shown if Figure 2.9. It is nearly spherical in form with an average diameter of approximately 20 mm, and is enclosed by three membranes. The front of the eye is covered by a transparent surface called the cornea. The remaining outer cover, called the sclera, is composed of a fibrous coat that surrounds the choroid, a layer containing blood capillaries. Inside the choroid is the retina, which is composed of two types of photo-receptors: cones and rods. The cones in each eye number between six and seven million. They are located primarily in the central portion of the retina, called fovea, and are highly sensitive to colour. Nerves connected to the retina leave the eyeball through the optic nerve bundle. Light entering the cornea is focused on the retinal surface by a lens which changes
FIG. 2.9 SIMPLIFIED CROSS SECTIONAL DIAGRAM OF THE HUMAN EYE
shape under muscular control to perform proper focusing of near and distant objects. The amount of light entering the eye is controlled by the iris. The central opening of the iris (the pupil) is variable in diameter from 2 mm up to 9 mm.

The number of rods is much larger, being of the order of 75 to 150 million distributed over the retinal surface. The rods are more sensitive to light than the cones, and function at low intensities [74]. The relative sensitivities of rods and cones as a function of an illuminating wave length are shown in Figure 2.10[77].

2.4.2 Brightness Adaptation and Discrimination

Since digital images are displayed as a discrete set of bright points, the ability of the eye to discriminate between different brightness levels is an important consideration in presenting image processing results. The range of light intensity levels to which the eye can adapt is enormous, being of the order of $10^{10}$ from the scotopic threshold (dim-light vision) to the glare limit [77]. There is also considerable experimental evidence which indicates that subjective brightness (i.e. brightness as perceived by the human visual system) is a logarithmic function of the light-intensity incident on the eye. This characteristic is illustrated in Figure 2.11 which is a plot of subjective brightness versus light intensity. The long solid curve represents the range
FIG. 2.10 SENSITIVITY OF CONES AND RODS (AFTER REF. 77)
of intensity to which the visual system can adapt. The transition from scotopic to photopic vision is gradual over the range from -3 to -1 on a log scale. The impressive dynamic range depicted in Figure 2.11 covered by changes in the eye's overall sensitivity is due to the phenomenon known as brightness adaptation. However, the total range of intensity levels it can discriminate simultaneously is rather small when compared with the total adaptation range. The current sensitivity level of the visual system is called the brightness-adaptation level which may correspond, for example, to brightness $B_a$ in Figure 2.11. The short intersecting curve represents the range of subjective brightness that the eye can perceive when adapted to this level. It is noted that this range is rather restricted, having a level $B_b$ at and below which all stimuli are perceived as indistinguishable blacks. The upper dashed portion of the short curve is not actually restricted but, if extended too far, loses its meaning because much higher intensities would simply raise the adaptation level to higher value than $B_a$.

The contrast sensitivity of the eye can be measured by exposing an observer to a uniform field of light of brightness $L$. The just noticeable difference in brightness $\Delta L$ increases proportionally with $L$. This is known as Weber's Law which states that

$$ \frac{\Delta L}{L} = K $$ (2.11)
FIG. 2.11  RANGE OF SUBJECTIVE BRIGHTNESS SENSATION SHOWING A PARTICULAR ADAPTATION LEVEL. (AFTER REF. 74)
where $K$ is a constant in the range 0.01 to 0.02. However, this result does not hold at very low and high values of $L^{[78]}$.

Weber's Law assumes that the observer's vision is adapted to the background luminance $L$. The eye adaptation mentioned above is an important mechanism which allows the observer a high contrast sensitivity over an enormous range of scene luminance. To allow for the viewing conditions Stevens$^{[78]}$ introduced a power relationship of the form

$$B = CL^\gamma$$

(2.12)

where $B$ is the absolute brightness, $C$ is a constant and $\gamma$ is an exponent which is adjusted to allow for viewing conditions. The value of $\gamma$ varies between $\frac{1}{2}$ and $\frac{1}{3}$ for light and dark surround conditions respectively.

A knowledge of $H(W)$, the sensitivity or spatial frequency response of the human visual response is useful in transform coding applications. The linear part of the model proposed by Mannos and Sakrison$^{[79]}$ is a filter transform function which indicates the relative sensitivity of the human visual system $H(W)$ to spatial light intensity distribution ($W$) as follows:

$$H(W) = 2.6 \left[ 0.0192 + 0.114 W \right] \exp \left\{ -(0.114W)^{1.1} \right\}$$

(2.13)
As depicted in Figure 2.12, $H(W)$ has a maximum value at $W = 8.0$ cycle/degree with a rapid decrease on either side.

![Graph showing the function $H(W)$ with peaks at $W = 8.0$ cycle/degree]

**FIG. 2.12** THE MANNOS AND SAKRISON HUMAN VISUAL SYSTEM RESPONSE (AFTER REF. 80)

In another approach Hall[81] has claimed that the incorporation of human visual system model in a Fourier transform coding scheme can improve the compression of still pictures by a factor of almost 10. He extended his techniques to code colour signals with good results at 1.0 bit/pel.
2.5 MEASURES OF PICTURE QUALITY

Image fidelity and intelligibility are commonly used to assess image quality. Image fidelity characterizes the departure of a processed version from some standard image, while image intelligibility denotes the ability of human eye or machine to extract relevant information from the image.

2.5.1 Image Fidelity Measures

In order to compare the results of such coding techniques a suitable error criterion must be defined. An error criterion which is often used is the percentage normalized mean square error (NMSE) is given by

\[
\text{NMSE} = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [X(m,n) - \hat{X}(m,n)]^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(m,n)^2} \cdot 100\% 
\]  

(2.14)

where \(X(m,n)\) and \(\hat{X}(m,n)\) are the original and processed image samples respectively, and \(MN\) is the total number of image samples. The other fidelity measure which can be used in image processing is the signal to noise ratio criterion given by

\[
\text{SNR} = 10 \log_{10} \frac{S_{PP}}{n_g} 
\]  

(2.15)
where $S_{pp}^2$ is peak to peak image signal power and $n_e^2$ is noise mean square error. The NMSE measure does not correlate well with the subjective evaluation of the image. However, it does provide an indication of the performance of an image processing system, and coupled with informal subjective testing, it is a useful tool for the measurement of image fidelity.

2.5.2 Subjective Rating-Scale Method for Image Quality

By using human observers, subjective rating of image quality has been developed by many workers [82-85]. Trained and untrained observers are used, where trained observers are better able to provide a critical judgement of picture quality.

There are two common types of subjective evaluation: absolute and comparative. In the absolute method, observers are shown an image and asked to judge its quality according to some predefined rating scale. The observer may be shown a set of reference images or may have to rely on his previous viewing experience. The comparative evaluation involves observer ranking of a set of images from 'best' to 'worst' for a particular group of images. Table II summarises the commonly used rating scales for subjective evaluation of images.
Table 2.II  Rating scales for subjective evaluation of images.

I  Quality Scale
  5  Excellent
  4  Good
  3  Fair
  2  Poor
  1  Bad

II  Impairment Scale
  5  Imperceptible
  4  Perceptible, but not annoying
  3  Slightly annoying
  2  Annoying
  1  Very annoying

III  Comparison Scale
  +3  Much better
  +2  Better
  +1  Slightly better
  0  The same
  -1  Slightly worse
  -2  Worse
  -3  Much worse

The subjective rating results are normally calculated as a mean opinion score defined as
\[
C = \frac{\sum_{i=1}^{k} n_i c_i}{K} \sum_{i=1}^{k} n_i
\]

where \( k \) is the categories number with associated category numbers \( c_1, c_2, c_3, \ldots, c_i, \ldots, c_k \) and \( n_i \) judgments are recorded in category number \( c_i \) for a particular impairment level. At least twenty subjects are considered necessary to ensure statistical confidence in such subjective image quality experiments.

The results of the subjective testing are influenced by the type of images presented to the viewer and the viewing conditions. If the images are familiar to the observer, he is more likely to be more critical of impairments because of preconceived notions of image structure. Further, test viewing conditions should be designed to match 'typical' viewing conditions as closely as possible. In the U.K. the viewing distance is specified as 6 times the picture height, the peak luminance as 50 cd/m², and the number of observers (who should be non-experts) around 20-25.
2.6 COMMENTS AND CONCLUSION

This chapter presents a summary of the major image-coding techniques which have been developed over the past twenty years or so. Given the recent rapid increases in computing power and speed, probably the most attractive modern technique is transform coding, the basic theory of which will be discussed in the next chapter.
CHAPTER III

THE THEORY
OF
ADAPTIVE TRANSFORM IMAGE CODING
3.1 INTRODUCTION

Transform theory has played a key role in image processing for a number of years, and it continues to be a topic of interest in theoretical as well as applied work in this field. Transform coding (in the past occasionally called block quantization) is a data compression technique in which a set of source elements is coded as a unit. The term "transform" indicates that the original set of elements is first processed by an invertible mathematical transformation prior to encoding. Figure 3.1 contains the basic block diagram representation of a transform-coding-decoding system. Several authors have analysed the theoretical concepts associated with transform coding [26-27],[31-32],[47-52].

For a correlated Gaussian source, optimum encoding consists of two steps; first, the decorrelation process, and then the application of a memoryless quantizer to the output of the transformation[19]. If the Gaussian assumption can be extended to image sources, then the optimum encoding should always first involve a decorrelation procedure. For a Gaussian source, decorrelation implies statistical independence[29], and the optimum transform for decorrelation under a mean square error criterion is usually identified as the K-L or Hotelling transformation [17]. In passing, it may be noted that the MSE criterion does not necessarily hold for systems in which the output is intended for the human eye.
FIG 3.1 BASIC TRANSFORM CODING SYSTEM.
The primary and obvious consideration is that transform coding introduces a distortion that is a function of the desired compression rate. An understanding of the trade-off between rate "R" and distortion "D" will lead to practical coding procedures, and the rate-versus-distortion formalism is fundamental to the proper design of a highly adaptive coding system.

Transform techniques are more complex than conventional and classical data compression algorithms, and may be either of nonadaptive or adaptive type. For a nonadaptive technique and a specified transmission rate (less than the original image rate), degradation of the image is essentially unavoidable. A non-adaptive algorithm is designed to operate identically for all images and all image regions. For such an algorithm, the image to be coded is assumed to come from a stationary source, and were the stationarity assumption valid, a non-adaptive coder could be an optimal image coder. In general the stationarity assumption is poor.

Images usually have varying statistical structures, both from image to image as well as within a single image. Nonstationarity results not only from scene nonstationarity but also from the imaging process itself\textsuperscript{[31]}. Nonadaptive transform coding is of little, if any, benefit compared with other conventional\textsuperscript{[32]} but simpler image-coding techniques\textsuperscript{[33]}. The real advantage comes with the use of adaptive methods.
In adaptive methods we are concerned with the operations of:
- sampling
- transformation
- sample selection
- quantization

and in an optimal adaptive system one would like all of the above processes to respond to variations in image statistics. Changing the parameters of each of the above operations, for example the sampling rate, the size or the type of transformation and the method of sample selection and quantization affects the overall performance of the encoder. Of course such techniques are not limited to transform coding systems and Figure 3.2 illustrates adaptive coding methods of various types.

3.2 IMAGE TRANSFORMS

As was indicated in the introduction, an original two-dimensional array of digital image elements, denoted by \( f(x,y) \), where \( x \) and \( y \) are the horizontal and vertical spatial co-ordinates, may be transformed either over the entire image or over sub-sections (blocks) of the image. The result of transforming a set of original data samples will be a corresponding set of transform coefficients denoted by \( T(m,n) \). In adaptive systems, the coefficients are adaptively selected and quantized according to the statistics of the original image data. These quantized samples are then encoded.
FIG. 3.2 ADAPTIVE CODING METHODS.
and transmitted through the channel. The samples at the receiver are decoded and then inverse transformed to reproduce the image. Figure 3.3 shows the general form of an adaptive transform coding system.

3.2.1 General Formulation and Representation

A one-dimensional, discrete transform can be expressed as a general relation:

\[ T(m) = \sum_{x=0}^{N-1} f(x) g(x,m) \quad m = 0,1,2,\ldots,N-1 \quad (3.1) \]

where \( T(m) \) are the transform coefficients
\( f(x) \) is the image vector
\( g(x,m) \) is the forward transform kernel

The process is represented in Figure 3.4.

The inverse transform is:

\[ f(x) = \sum_{m=0}^{N-1} T(m) h(x,m) \quad ; \quad x = 0,1,2,\ldots,N-1 \quad (3.2) \]

where \( h(x,m) \) is the inverse transform kernel. For a two-dimensional transform the forward and inverse transforms are given by the relations:

\[ T(m,n) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) g(x,y,m,n) \quad (3.3) \]
FIG. 3.3 ADAPTIVE TRANSFORM CODING SYSTEM.
FIG. 3.4 GENERALIZED ONE-DIMENSIONAL TRANSFORMATION.
and

\[ f(x,y) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} T(m,n) h(x,y,m,n) \]  

(3.4)

respectively, where \( g(x,y,m,n) \) and \( h(x,y,m,n) \) are the forward and inverse kernels. It is represented in Figure 3.5.

The forward kernel is said to be separable if:

\[ g(x,y,n,m) = g_1(x,n) \cdot g_2(y,m) \]  

(3.5)

The kernel is in addition symmetric, (as for example, the Fourier transform), if \( g_1 \) is functionally equal to \( g_2 \). In this case, Equation 3.5 can be expressed in the form:

\[ g(x,y,n,m) = g_1(x,n) \cdot g_1(y,m) \]  

(3.6)

Identical comments hold for the inverse kernel if \( g(x,y,n,m) \) is replaced by \( h(x,y,n,m) \) in Equations (3.5) and (3.6).

To compute a two-dimensional transform with a separable kernel the process can be divided into two steps[74], requiring a one-dimensional transform for each step. First the one-dimensional transform can be taken along each row of \( f(x,y) \) (i.e. \( y \) constant) yielding:

\[ b(y,n) = \sum_{x=0}^{N-1} f(x,y) g_1(x,n) \]  

(3.7)
FIG. 3.5 GENERALIZED TWO-DIMENSIONAL TRANSFORMATION.
for \( n, y = 0, 1, 2, \ldots, N-1 \). Next a one-dimensional transform is taken along each column of \( B(y,n) \) (i.e. \( n \) constant) resulting in the expression:

\[
T(m,n) = \sum_{y=0}^{N-1} B(y,n) g_2(y,m)
\]  

(3.8a)

For \( m, n = 0, 1, 2, \ldots, N-1 \).

Similar comments hold for the inverse transform if \( h(x,y,m,n) \) is separable.

The operation above is illustrated in Figure 3.6.

In the three-dimensional case for an image array \( f(x,y,z) \) where \( x, y, \) and \( z \) are the horizontal, vertical and temporal co-ordinates the forward and inverse transforms are given by the equations:

\[
T(m,n,t) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sum_{z=0}^{T-1} f(x,y,z) g_1(x,m) g_2(y,n) g_3(z,t)
\]  

(3.8b)

\[
f(x,y,z) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} T(m,n,t) h_1(x,m) h_2(y,n) h_3(z,t)
\]  

(3.8c)

It is often useful to express transforms in vector-space or matrix form\[^{75}\]. Thus let \( F \) and \( f \) denote the matrix and vector forms of the image array, and let \( A \) and
FIG. 3.6 ILLUSTRATES COMPUTATION OF A TWO-DIMENSIONAL TRANSFORM WITH A SEPARABLE KERNEL.
a be the matrix and vector forms of the transformed image. Then the transform written in vector form is given by

$$\mathbf{a} = \mathbf{Gf} \quad (3.9)$$

where $\mathbf{G}$ is the forward transform matrix. The inverse transform is

$$\mathbf{f} = \mathbf{Ha} \quad (3.10)$$

where $\mathbf{H}$ is the reverse transform matrix. Hence,

$$\mathbf{H} = \mathbf{G}^{-1} \quad (3.11)$$

If $\mathbf{G}$ is a unitary matrix, then by definition

$$\mathbf{G}^{-1} = (\mathbf{G}^*)^T \quad (3.12)$$

where $\mathbf{G}^*$ is the complex conjugate matrix of $\mathbf{G}$ and where $\mathbf{G}^T$ is the matrix transpose of $\mathbf{G}$. A real, unitary matrix is called an orthogonal matrix. For such a matrix

$$\mathbf{G}^{-1} = \mathbf{G}^T \quad (3.13)$$

In addition, if $\mathbf{G}$ is a symmetric orthogonal matrix, which is often the case for commonly used image transforms, then

$$\mathbf{G}^{-1} = \mathbf{G} \quad (3.14)$$

If the transform kernel is separable such that

$$\mathbf{G} = \mathbf{G}_c \times \mathbf{G}_r \quad (3.15)$$

where $\mathbf{G}_c$ and $\mathbf{G}_r$ are unitary matrices representing the column and row transformations, then the transformed image
matrix, $A$, can be obtained from the image matrix, $F$, by

$$A = G_C F G_R^T$$  \hspace{1cm} (3.16)

The inverse transformation is given by

$$F = H_C A H_R^T$$  \hspace{1cm} (3.17)

where $H_C = G_C^{-1}$ and $H_R = G_R^{-1}$

3.2.2 Types of Image Transforms

3.2.2.1 The Fourier Transform

The discrete Fourier transform has been used extensively in signal analysis and for image coding. The two-dimensional Fourier transform of an image $f(x,y)$ may be expressed in series form as

$$T(m,n) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-\frac{j2\pi}{N} (mx + ny)\right]$$  \hspace{1cm} (3.18)

for $m,n = 0,1,2 \ldots N-1$, and its inverse as

$$f(x,y) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} T(m,n) \exp\left[\frac{j2\pi}{N}(mx + ny)\right]$$  \hspace{1cm} (3.19)

for $x,y = 0,1,2 \ldots N-1$; where $j = \sqrt{-1}$.

The basis functions of the transform are complex
exponentials and may be decomposed into sine and cosine components, i.e.

\[ A(x,y ; m,n) = \exp \left[ -\frac{12\pi}{N} (mx + ny) \right] \]

\[ = \cos \left[ \frac{2\pi}{N} (mx + ny) \right] - j \sin \left[ \frac{2\pi}{N} (mx + ny) \right] \]

\[ B(x,y ; m,n) = \exp \left[ \frac{12\pi}{N} (mx + ny) \right] \]

\[ = \cos \left[ \frac{2\pi}{N} (mx + ny) \right] + j \sin \left[ \frac{2\pi}{N} (mx + ny) \right] \]

(3.19a)

(3.19b)

Figure 3.7 contains the plots of the sine and cosine components of the one-dimensional Fourier basis function for \( N = 16 \).

The spectral component at the origin of the Fourier domain

\[ T(0,0) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \] (3.19c)

is equal to \( N \) times the spatial average of the image plane intensity. If \( f(x,y) \) represents a luminance field, then \( T(0,0) \) is the average brightness in the spatial domain.
FIG. 3.7 FOURIER BASIS FUNCTIONS FOR N = 16
The two-dimensional Fourier transform of an image is essentially a Fourier series representation of a two-dimensional field. For the Fourier series representation to be valid, the field must be periodic. For a real positive function, \( f(x,y) \), its Fourier transform will, in general, be complex containing \( 2N^2 \) components. However, \( T(m,n) \) exhibits the property of conjugate symmetry, and therefore one-half of the transform domain samples are redundant, that is, they can be generated from the other transform samples.

The transformation matrix \( A \) can be written in direct product form as

\[
A = A_C \otimes A_R
\]

where

\[
A_R = A_C = \frac{1}{\sqrt{N}}
\]

\[
\begin{array}{cccccccc}
    w^0 & w^0 & w^0 & \ldots & \ldots & w^0 \\
    w^0 & w^1 & w^2 & \ldots & w^{N-1} \\
    w^0 & w^2 & w^4 & \ldots & w^{2(N-1)} \\
    \cdot & \cdot & \cdot & \cdot & \cdot \\
    \cdot & \cdot & \cdot & \cdot & \cdot \\
    \cdot & \cdot & \cdot & \cdot & \cdot \\
    w^0 & \ldots & \ldots & \ldots & w^{(N-1)^2} \\
\end{array}
\]

with \( W = \exp\left[-2\pi j/N\right] \)
3.2.2.2 The Hadamard Transform

The Hadamard Transform [34] is based on the Hadamard matrix, which is a square array of plus and minus ones whose rows and columns are orthogonal. This transform is convenient to use because of its computational simplicity - the generation of the transform coefficients involving additions and subtractions only. Note from Figure 3.8 that the zero crossings are not always equally spaced. This leads to the concept of sequency defined as one-half of the average number of zero crossings per unit interval along each row of the Hadamard matrix.

The two-dimensional Hadamard transform pair sequency can be expressed by the equations:

\[
H(m,n) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)(-1)^{m+n} 
\]

where \( N = 2^n \), \( b_i(x) \) is the \( i \)th bit in the binary representation of \( i \) and

\[
f(x,y) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} H(m,n)(-1)^{m+n} 
\]

(3.20)
Since the forward and inverse transforms are identical, an algorithm used for computing $H(m,n)$ can be used without modification to obtain $f(x,y)$ and vice versa.

A normalized $N \times N$ Hadamard matrix, $H$, satisfies the relation

$$H H^T = I$$

(3.22)

where $I$ is the identity matrix.

The Hadamard matrix of lowest order (i.e. $N=2$) is given by:

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

(3.23)

and the construction of a Hadamard matrix of order $N$ can be carried out by using the following recursive relationship:

$$H_N = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{bmatrix}$$

(3.24)

where $N = 2^n$ and $n$ is integer. Since the Hadamard matrix is real, symmetric and orthonormal, the forward and inverse transforms can be expressed as:

$$A = HFH$$

(3.25a)

$$F = HAH$$

(3.25b)
As a further example, the Hadamard matrix of order four is given by

\[ H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & H_2 \end{bmatrix} = \begin{bmatrix} + & + & + \\ + & - & - \\ + & + & - \\ + & - & + \end{bmatrix} \] (3.26)

where + and - indicate +1 and -1 respectively.

Figure 3.8 illustrates the Hadamard transform basis functions for \( N = 16 \).

### 3.2.2.3 The Karhunen-Loeve (Hotelling) Transform

The K-L Transformation is a special case of an eigenvector matrix\[35\text{-}37\]. It takes the general form

\[
T(m,n) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) g(x,y,m,n) \tag{3.27}
\]

for which the kernel \( g(x,y,m,n) \) satisfies the equation,

\[
\lambda(m,n) g(x,y,m,n) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} K_f(x,y,x',y') g(x',y',m,n) \tag{3.28}
\]

where

\(\lambda\) is a constant for fixed \((m,n)\)

\(K_f(x,y,x',y')\) denotes the covariance function of the image array, and is defined as
FIG. 3.8 HADAMARD TRANSFORM BASIS FUNCTIONS FOR N = 16
\[ K_f (x,y,x',y') = E \{ [f(x,y) - E\{ f(x,y) \}] [f(x',y') - E\{ f(x',y') \}]^T \} \]  

(3.29)

where \( E(.) \) denotes the expected value operator. The set of functions defined by the kernel, \( g(x,y,m,n) \), are the eigenvectors of the covariance function, and \( \lambda(m,n) \) represents the eigenvalues of the covariance function.

In matrix form, the transformation matrix \( G \) satisfies the relation

\[ G K_f = \lambda G \]  

(3.30)

where

- \( K_f \) is the covariance matrix of \( f \)
- \( G \) is a matrix whose rows are eigenvectors of \( K_f \)
- \( \lambda \) is a diagonal matrix of the form:

\[
\lambda = \begin{bmatrix}
\lambda(1) & 0 & \cdots & 0 \\
0 & \lambda(2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & \lambda(N^2)
\end{bmatrix}
\]

(3.31)

Generally, \( K_f \) is not separable. Figure 3.9 contains plots of the (K-L) basis function for a one-dimensional Markov process with adjacent element correlation of 0.9.
FIG. 3.9 K-L TRANSFORM BASIS FUNCTIONS FOR N = 16
3.2.2.4 The Haar Transform

The Haar transform is derived from the Haar matrix\cite{48}, which is a square array containing a number of zero elements. Examples of 4 x 4 and 8 x 8 orthonormal Haar matrices are shown below:

\[
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
\sqrt{2} & -\sqrt{2} & 0 & 0 \\
0 & 0 & \sqrt{2} & -\sqrt{2}
\end{bmatrix}
\] (3.32a)

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
\sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2}
\end{bmatrix}
\] (3.32b)

Based on these patterns, the Haar matrices of order \( N = 2^N; n = 1, 2, 3, \ldots \), can be constructed. Haar basis functions for \( N = 16 \) are shown in Figure 3.10. Unlike other transforms (in which the transform coefficients are the weighted sums of all the data sample values), the form of the Haar matrix implies that some of the transform coefficients are functions of only some selected input
FIG. 3.10 HAAR TRANSFORM BASIS FUNCTIONS FOR N = 16
values, due to the presence of many zeros in the matrix\cite{2}.

The two-dimensional Haar transform in matrix form is

\[ A = R F R^T \]

and its inverse is

\[ F = R^T A R \]

where \( A \) and \( F \) are the transform and image arrays, and \( R \) and \( R^T \) are, respectively, the Haar matrix and its transpose.

3.2.2.5 The Slant Transform

The Slant transform\cite{38-39} has the following properties:

1. A constant dc basis vector
2. Sequency property (the number of zero crossings increases with increasing coefficient order)
3. Slant basis vectors (monotonically decreasing in constant steps from maximum to minimum amplitude).

The Slant transform matrices of order 2 and 4 are:

\[
[S_2] = \frac{1}{\sqrt{2}} \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix}
\] (3.35)
Slant transform basis functions are shown in Figure 3.11 for $N = 16$.

3.2.2.6 The Discrete Cosine Transform (DCT)

The performance of the Cosine Transform is virtually identical to that of the K-L transform for numerous practical applications in which the Markov type source assumption is valid\cite{27} and it has been shown\cite{3} that the DCT is the KLT for Markov image data in the limit as $\rho \to 1$, where $\rho$ is the interelement correlation coefficient.

The cosine transform, unlike the K-L transform, can be implemented computationally through a "fast" algorithm, and approaches the K-L transform in efficiency with neither knowledge nor utilization of the source correlation. For the K-L transform, the source covariance model must be available to derive the actual transform matrix.

The cosine transform is a strictly deterministic transform. Conversely, the K-L "transform" is a class of transformation, and for each application is a function of the appropriate
Fig. 3.11 Slant transform basis functions for \( n = 16 \)
covariance matrix. It is a significant practical benefit that the single deterministic transformation closely approaches in performance the entire class of theoretically optimum transformations.

The discrete cosine transform (DCT) of a data sequence \( f(x), x = 0, 1, 2, \ldots, N-1 \) is defined as \(^{[41]}\)

\[
T(0) = \frac{\sqrt{2}}{N} \sum_{x=0}^{N-1} f(x) \quad (3.37)
\]

\[
T(m) = \frac{2}{N} \sum_{x=0}^{N-1} f(x) \cos \left[ \frac{m \pi}{N} \left( x + \frac{1}{2} \right) \right] \quad (3.38)
\]

for \( m = 1, 2, \ldots, N-1 \)

The inverse cosine transform is given by:

\[
f(x) = \frac{1}{\sqrt{2}} T(0) + \sum_{m=0}^{N-1} T(m) \cos \left[ \frac{m \pi}{N} \left( x + \frac{1}{2} \right) \right] \quad (3.39)
\]

The DCT for the two-dimensional case given a square image array \( f(x,y), x,y = 0, 1, 2, \ldots, N-1 \) is

\[
T(m,n) = \frac{2}{N} C(m) C(n) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x,y) \cos \left[ \frac{m \pi}{N} \left( x + \frac{1}{2} \right) \right] \cdot \cos \left[ \frac{n \pi}{N} \left( y + \frac{1}{2} \right) \right]
\]

\[(3.40)\]
The inverse cosine transform (IDCT) is

\[ f(x,y) = \frac{2}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} C(m)C(n)T(m,n) \cos\left(\frac{m\pi}{N}(x + \frac{1}{2})\right) \cos\left(\frac{n\pi}{N}(y + \frac{1}{2})\right) \]

(3.41)

where \( C(0) = \frac{1}{\sqrt{2}} \), and \( C(1), C(2), \ldots, C(N-1) = 1 \)

The basis functions of the cosine transform are a class of Chebycheff polynomials [41]. Figure 3.12 shows the cosine transform basis functions of order 16.

From Figure 3.12 and Equation (3.40) the efficiency of the DCT in terms of energy packing can be demonstrated. Thus a close resemblance between basis vectors of the DCT and the signal vectors results in large magnitude DCT coefficients. A transform, such as the DCT, packs most energy into the low sequency coefficients, implying that the low sequency basis vectors of a good image transform resembles slowly varying image signal vectors. For example, as shown in Figure 3.12, the low sequency basis vectors of the 16 x 16 DCT change smoothly whilst other transform basis vectors (for example Walsh and Haar) have sudden jumps between the positive and negative elements.
FIG. 3.12 DISCRETE COSINE TRANSFORM BASIS FUNCTIONS FOR N = 16
3.3  OPTIMISATION OF PARAMETERS

In transform coding, data compression is essentially achieved by two processes. The first is the transformation which packs most of the signal energy into a few coefficients. The second is the quantization process in which the quantization error should be kept to a minimum. Therefore, to obtain efficient transform coding schemes which, whilst remaining relatively simple in implementation, achieve significant reduction in bit rate, both processes have to be optimum. Thus one has to choose the right transform and the right block size to optimize the first process; and the right quantizer and allocate a proper number of bits to each coefficient to optimize the second. The following two sections are devoted to some of these considerations.

3.3.1 Selection of Transform

The transformation that offers the best performance is the KLT which has the best energy packing performance, and ability to decorrelate signal data, and results in the least mean square error. However, the KLT is data dependent and necessitates the computation of eigenvectors which creates many problems and prevents it from being widely used.

In practice, the choice of transformation lies very much between the Hadamard transform and the DCT, depending on whether or not processing speed is paramount. The discrete cosine transform has a performance close to that of the
KLT and therefore it is frequently used. The computational efficiency of the fast algorithms for the discrete Fourier transform[11], Hadamard transform[13], Haar transform[15], Slant transform[39], and discrete cosine transform[41] are shown in Table 3.I for the one-dimensional case.

Table 3.I Computational Requirement of Image Transforms

<table>
<thead>
<tr>
<th>Transform</th>
<th>Computational Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier</td>
<td>((N \log_2 N)) Complex additions and multiplications</td>
</tr>
<tr>
<td>Hadamard</td>
<td>((N \log_2 N)) additions</td>
</tr>
<tr>
<td>Haar</td>
<td>(2(N-1)) additions</td>
</tr>
<tr>
<td>Slant</td>
<td>((N \log_2 N) + 2N - 4) additions and multiplications</td>
</tr>
<tr>
<td>Cosine</td>
<td>(\frac{3N}{2} \left(\log_2 N - 1\right) + 2) addition and ((N \log_2 N) - \frac{3N}{2} + 4) multiplications</td>
</tr>
<tr>
<td>KL</td>
<td>No fast algorithm</td>
</tr>
</tbody>
</table>

3.3.2 Selection of Block Size

Mean-square error performance should improve with increasing block size \((n)\), since the number of correlations considered also increase. However, most pictures contain
significant correlation between elements for only about 20 adjacent pels, although this number depends strongly on the amount of activity in the picture. Therefore, it seems that very little can be gained by using block sizes larger than 32.

This can be demonstrated in Figure 3.13 which contains a plot of the mean square error for an image with a Markov process covariance as a function of block size for various transformations[21]. In this plot the 25 percent of coefficients with the largest variances were selected, and the remainder set to zero. From the Figure it is seen that the rate of decrease in mean square error for large block sizes becomes quite small for sizes larger than 16 x 16. In addition, large block sizes introduce two distinct disadvantages. These are the requirement for a large buffer memory and the difficulty in achieving adaptation within a block. It should be noted that the choice of block size also depends on the kind of coding schemes used. In general, an optimum block size will be between 4 and 32.

3.4 PROBABILITY DENSITY MODEL FOR IMAGE TRANSFORMS

Generally the probability density function (pdf) of the transform coefficients is difficult to obtain since the short-term statistics of image are not well defined. The first coefficient T(0,0) is the weighted sum of original pels x(i,j) which are non-negative. As such T(0,0) itself
FIG. 3.13 MEAN SQUARE ERROR PERFORMANCE OF VARIOUS IMAGE TRANSFORMS AS A FUNCTION OF BLOCK SIZE FOR A TWO-DIMENSIONAL MARKOV SOURCE WITH HORIZONTAL AND VERTICAL CORRELATION FACTORS OF 0.95 (AFTER REF. 21)
is non-negative and its pdf is occasionally modelled by a Rayleigh distribution given by:

\[ P(T(0,0)) = \frac{T(0,0)}{\sigma_T(0,0)} \exp \left[ - \frac{T^2(0,0)}{2\sigma_T^2(0,0)} \right] \] (3.42)

where \( \sigma_T(0,0) \) is the coefficient variance.

For other coefficients \( T(u,v) \) where \( (u,v) \neq 0 \), Gaussian or Laplacian pdfs have been used. The Gaussian pdf is given by:

\[ P(T(u,v)) = \frac{1}{\sqrt{2\pi\sigma_T^2(u,v)}} \exp \left[ - \frac{T^2(u,v)}{2\sigma_T^2(u,v)} \right] \] (3.43)

and the Laplacian pdf by:

\[ P(T(u,v)) = \frac{1}{\sqrt{2 \sigma_T(u,v)}} \exp \left[ - \frac{2|T(u,v)|}{\sigma_T(u,v)} \right] \] (3.44)

The use of other models, such as the Gamma distribution [92] has also been reported.

3.5 COMMENTS AND CONCLUSIONS

The primary purpose of the transformation is to convert statistically dependent picture elements into more independent coefficients\[21\], which can be quantized and coded for the purpose of transmission over the channel at a low bit-rate.
and with minimum mean-square error. It is therefore desirable to have a transform which compacts most of the image energy into as few coefficients as possible. Another consideration is the ease of performing the transformation itself.

The K-L transform is the optimum transform, but there are problems with its use in practice. First of all, the covariance function of an image is not stationary, and therefore, one must either choose different covariance matrices matched to different regions of the picture or use an average (with a consequent loss in performance). The second problem is in the computation of the eigenvectors of the covariance matrix, and the difficulty of actually implementing the transform.

The obvious difference between the discrete Fourier transform (DFT), discrete cosine transform (DCT), Hadamard transform, Slant transform and the Karhunen-Loeve transform (KLT) [21] is that all the former are independent of image statistics. Another difference is in the ease of implementation. All the above transforms are unitary (see Appendix I) and therefore inverse transformation at the receiver is as easy to implement as the original transformation itself. It is interesting to observe that for small sub-pictures the mean-square error performance of different transforms is similar to that of the KLT. As the block size increases the Hadamard transform does not perform as well, but the
Fourier transform improves. The discrete cosine transform remains close to the optimum for all block sizes. Figures 3.9 and 3.12 show the similarity of basis functions of the KLT and DCT. Due to this similarity the DCT and KLT are very close to each other in performance. It can be proved that for the first-order Markov process with exponential correlation, for most usually occurring values of correlation coefficient the best fast* transform is the discrete cosine transform. It should be remembered, however, that the above remarks are strictly valid only for data from a stationary Markov process with exponential correlation. For real images these assumptions are not valid, particularly if the block size is small, and therefore simulations using unitary transforms in general show results which in many cases, are quite inferior to those obtained using the KLT [21]. The best all-round choice is probably the discrete cosine transform.

* - one which can be computed by an algorithm similar to the FFT.
CHAPTER IV

ADAPTIVE TRANSFORM CODING TECHNIQUES
4.1 INTRODUCTION

Adaptive transform coding techniques are more efficient than non-adaptive schemes since the parameters of a transform coder can be matched to the statistics of the subpicture being coded. As the latter may be highly non-stationary, adaptation can increase the coding efficiency significantly. Two types of adaptation are possible:

a) Where changes in the parameters of the transform coder are based on the previously transmitted data.

b) Where some future data is used to compute the parameter changes of the transform coder.

Adaptive transform coding schemes may consist of a combination of some or all of the following techniques:

i. Adaptive sampling, which gives better resolution for areas with greater image activity and therefore reduces block effects.

ii. Adaptive coefficient selection, which gives better performance mainly due to the fact that coefficients are selected adaptively depending on criteria such as variance, energy, magnitude, etc., with the hope of reducing the overall normalized mean square error (NMSE). This in turn will again give reduced block effects and better subjective performance.
iii. Adaptive quantization, which gives a good signal to noise ratio (SNR) due to low quantization noise.

iv. Adaptive bit assignment - bits are variously assigned to the selected coefficients based on the rate distortion function. For low overall distortion more bits are assigned to the higher amplitude coefficients. If a Huffman code is used, then the overall bit rate will be lower than that of non-adaptive schemes.

Two advantages of adaptive transform coding schemes are:

1. Greater coding efficiency - a lower bit rate mainly because of adaptation to the statistics of the transform coefficients.

2. Better subjective quality - higher signal to noise ratio (SNR) values and reduction in block effects.

4.2 **ADAPTIVE COEFFICIENT SELECTION AND QUANTIZATION - A REVIEW**

To achieve an adaptive transform scheme the coefficients must be adaptively selected and quantized. Figure 4.1(a) shows a block diagram of an adaptive transform coding scheme, where the transform coefficients have been coded depending on the short-term statistical characteristics of the image. Such a scheme normally results in a variable rate data stream, and a buffer will be needed for operation in practice. In general, the more established techniques that have been employed are:
FIG. 4.1(a) AN ADAPTIVE TRANSFORM CODING SCHEME.
1. Threshold sampling
2. Zonal sampling

4.2.1 Threshold Sampling

This method involves setting a threshold level and then transmitting only transform coefficients which are larger than the threshold. The transform coefficients below the threshold level are set to zero at the receiver. This system is adaptive because the number and locations of the selected coefficients are changing in the transform domain from one block to another depending upon picture detail. In this system addressing information must be transmitted to indicate the location of the retained coefficients, which requires a relatively high bit rate.

An adaptive threshold coding method was proposed by Anderson and Huang,[54] using a two-dimensional Fourier transform and a block size of 16 x 16 picture elements. The standard deviation of the coefficients in each block was measured, and the amplitudes, phases, and positions of the M transform coefficients with the largest amplitudes transmitted, where M was proportional to the standard deviation of the coefficients in each block. The adaptivity of the system was increased by making the number of quantization levels in each block proportional to the standard deviation of the transform coefficients in that block. The addressing information and position of the
coefficients retained were transmitted using a run-length coding algorithm. Good results were reported at 1.25 bits per picture element.

Reader[64] used adaptive zonal and threshold coding techniques within a transformed block together with the Slant transform. Zonal coding was used inside the maximum variance zone, and threshold coding outside the defined zone in a particular block. It is considered that the result obtained (a total of 1.5 bits per element) is not a significant improvement as compared to simple zonal coding, and does not justify the effort involved (addressing information for those threshold coded coefficients outside the maximum variance zone causes an increase in the average bit rate). Recently, a threshold adaptive coder has been described [134] using the discrete cosine transform (DCT). In this technique an initial threshold is established, and those transform coefficients whose magnitudes are greater than the threshold have the threshold value subtracted.

Thus

\[
T_{th}(u,v) = \begin{cases} 
T(u,v) - th & \text{if } T(u,v) > th \\ 
0 & \text{if } T(u,v) \leq th 
\end{cases}
\]  

(4.1a)

where th is the threshold value. The value of the threshold varies with respect to the desired bit rate. The coefficients \(T_{th}(u,v)\)
are then scaled by a feedback normalization factor $D(M)$ defined by the output rate buffer according to the relation

$$T_{th,N}(u,v) = \frac{T_{th}(u,v)}{D(M)}$$  \hspace{1cm} (4.1b)

where $M$ is the number of the block (of size $16 \times 16$ elements). The scaled coefficients are quantized using a floating point to integer roundoff conversion given by

$$\hat{T}_{th,N}(u,v) = \text{Integer part of } [T_{th,N}(u,v) + 0.5]$$  \hspace{1cm} (4.1c)

and then Huffman coded and fed to the rate buffer. The buffer status and input rate are monitored to generate the coefficient scaling factor. At the receiver, the received fixed rate data is fed to a rate buffer that generates Huffman code words at a variable rate for decoding. The decoded transform coefficients are inverse normalized by the feedback parameter, added to the threshold, and inverse transformed to reconstruct the image. Good results are reported for real-time colour coding at 0.4 bits per element!

### 4.2.2 Zonal Sampling

In the transform domain most of the transformed image energy is concentrated in relatively few low order coefficients. This results from the high spatial correlation
between the original image elements, and substantial bandwidth reduction may be accomplished if the higher order coefficients are not transmitted. Discarding the higher order coefficients is equivalent to passing the image through a zonal low-pass filter. Several different zone shapes may be used, for example, rectangular, triangular, elliptical or circular. However, it may be shown that the optimum zone for a mean square error criterion is one of maximum variance, where the transform coefficients having the largest variances for a given covariance model of the original image are selected. This zonal sampling operation may be expressed analytically as:

$$\hat{\mathbf{f}} = (G^{-1} S^T)(S G)\mathbf{f}$$

(4.2)

where $S$ is the selection matrix, $G$ the image transform, $\mathbf{f}$ the original data vector and $\hat{\mathbf{f}}$ the reconstructed data vector. $(S G)$ represents the forward zonal sampling process and $(G^{-1} S^T)$ represents the inverse reconstruction process. Figure 4.1(b) contains a block diagram of the zonal sampling technique, where $\mathbf{a}$ represents the transformed vector, $\mathbf{a}^T$ its transpose, and $\hat{\mathbf{a}}$ is the reconstructed zonal vector.

4.2.3 Zonal Coding

In the zonal transform coding system a zone or set of zones is established in each transform block. The zones may be derived from an optimum block quantization algorithm where the number of quantization levels for each transform
FIG. 4.1(b) ZONAL SAMPLING OPERATION.
coefficient is set proportional to its expected variance. The quantization error for each transform coefficient is the same\[26\], because although the variances of the transform coefficients are different, the number of binary digits assigned to each coefficient will be different also. In addition, efficient encoding requires more binary digits for areas of high detail and fewer binary digits for areas of lower detail. If a constant word-length code is used, \( b(m,n) \) code bits are assigned to coefficient \( T(m,n) \) resulting in a total of \( B \) bits per block, i.e.

\[
B = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} b(m,n) \text{ bits} \tag{4.3}
\]

Huang and Schultheiss\[19\] determined the optimum allocation of a total of \( B \) bits to the \( N^2 \) coefficients. They found that the number of bits \( b(m,n) \) to code coefficient \( T(m,n) \) should be proportional to \( \sigma^2_T(m,n) \). An algorithm was obtained for computing \( b(m,n) \) such that the mean square quantization error was minimised for a given \( B \) and set of variances \( \sigma^2_T(m,n); m, n = 0, 1 \ldots N-1. \)

### 4.2.4 Other Adaptive Transform Coding Schemes

An adaptive Fourier transform quantization technique in which the magnitude and phase of each coefficient are adaptively coded has been developed by Tescher et al.\[61\]
In this technique the number of quantization levels for the phase and magnitude components are set proportional to the logarithm of their estimated variances, under the condition that the number of phase levels is twice the number of magnitude levels. The variance of the amplitudes of the individual coefficients is estimated using a predictor that combines the variances of a number of adjacent quantized elements. This adaptive coding technique has also been extended to the coding of Hadamard transform coefficients.

In another approach Tescher and Cox [62] have reported good results using a recursive quantization method with cosine and slant transforms. In this method the image is divided into blocks of 16 x 16 pels and the transform coefficients scanned diagonally. Then an estimate of the variance of the one-dimensional coefficient sequence is made and a bit assignment set proportional to its logarithm. The estimate of the variance of the $n$th transform coefficient $\sigma_{T^2}(n)$ is

$$\sigma_{T^2}(n) = B_1 \sigma_{T^2}(n-1) + (1 - B_1) \hat{T}^2(n-1) \quad (4.4)$$

where $B_1$ is a weighting factor which was chosen as 0.75 in their experiments, and $\hat{T}(n-1)$ is the quantized amplitude of the $(n-1)$th transform coefficient (in the one-dimensional sequence).
Another adaptive coding system has been proposed by Mauersberger[65], in which a two-dimensional variable relating to the mean square difference between adjacent pels has been used to provide estimates of the horizontal and vertical correlation which are subsequently used to classify the block, based on the rate distortion criterion.

4.2.5 Adaptive Transform Coding Methods Using an Activity Index

Another approach to adaptation is to construct a measure of spatial activity in the subpicture and then to adapt the coefficient selection accordingly. This will then acknowledge the lower sensitivity of the eye to amplitude variations in regions of high spatial detail. Gimlett[55] and Claire [56] have proposed definitions of "activity index" to classify transform blocks, where in each class a different coefficient selection and quantization procedure is used. Two definitions have been used, the first being the sum of the squares of the amplitudes of the transform coefficients and the second the sum of their absolute amplitudes. A combination of zonal and threshold sampling is used for each class. Tasto and Wintz[57], classified a (6 x6) block into one of three categories, based on block statistics. For each category an average covariance matrix and corresponding set of eigenvectors are used for transforming the picture elements of the subpicture. The overhead information for adaptation is only 2 bits per block,
and improvements in coding efficiency of as much as 30 to 50 per cent over the nonadaptive case are reported. Tescher et al. [58, 59], estimated the variance of the transform coefficients and then assigned bits in proportion to the logarithm of the estimated variance. Overall this results in assignment of more bits for transform blocks with large coefficients, and fewer bits for those having small magnitude coefficients.

An efficient adaptive encoding scheme using the DCT has been developed by Chen and Smith[63] which uses an activity index to classify transform blocks. In this scheme adaptivity results from a distribution of bits between classes, favouring those with higher levels of activity. The coded blocks are sorted into four classes according to level of image activity measured by the total a.c energy within each transform block. Good results are reported for both noiseless and noisy transmission at a total rate (including overhead information) of 1 bit and 0.5 bit per element respectively for monochrome images and for a total rate of 2 bits and 1 bit per element for colour images. Adaptive coding techniques result in a variable data rate where the number of bits assigned to each block changes from block to block, and therefore a buffer and buffer control logic are needed for transmission over a fixed rate channel. This matter is discussed further in Chapter V.
Sections 4.2.4 and 4.2.5 presented a brief review of a variety of adaptive transform coding algorithms. In general, the bit rates which are achievable using such systems are in the region of $2^{-1}$, and $1.5 - 0.5$ bits per element for the one and two-dimensional cases respectively. These figures represent a substantial reduction in information rate over nonadaptive systems. However, there are still some areas which have been insufficiently explored and in the following section some improvements are described which enhance the performance and reduce the complexity of adaptive transform coding systems.

Chen and Smith's basic technique has been chosen for an investigation of adaptive transform coding techniques in order to develop a new energy estimation method for the purpose of reducing the number of calculations. Satisfactory results are obtained both for this and for the total bit rate for successful image reconstruction. Mean square error between original and reconstructed images is used as the primary performance criterion, and subjective judgements made by visual inspection of the image coding examples presented.

4.3 ADAPTIVE TRANSFORM CODING USING THE DCT AND ACTIVITY INDEX CLASSIFICATION VIA A SIMPLIFIED ENERGY ESTIMATION METHOD

4.3.1 Introduction

In this section an efficient adaptive transform coding
technique using the discrete cosine transform (DCT) for bandwidth compression of monochrome images is investigated. The aim is to achieve a satisfactory balance between complexity of implementation and performance, and also computational simplification, through an alternative energy estimation method, and Figure 4.2 shows the block diagram of a typical adaptive transform coding system using an activity index.

Adaptive transform coding based upon the statistics of image data has proved to be very effective but involves great system complexity, and a practical approach is to select an efficient coding scheme which achieves a successful compromise between complexity and performance, for example, the DCT together with activity index classification of the block i.e. the transform blocks are sorted into different classes by differing levels of image activity class. Coding bits are then allocated to individual transform coefficients according to the variance matrix of the transformed data. Thus bits are distributed between the higher and lower classes, favouring higher levels of activity over lower levels, to provide the desired adaptivity. The transform coefficients are non-uniformly quantized, and the mean square error between the original and reconstructed images computed as a performance criterion. Direct visual comparison between the original and reconstructed images has also been carried out. Good results are achieved in terms of both the overall bit rate and the simplicity of the system.
FIG. 4.2 A TYPICAL BLOCK DIAGRAM OF AN ADAPTIVE TRANSFORM
IMAGE CODING SYSTEM USING AN ACTIVITY INDEX
4.3.2 The Cosine Transform

The one-dimensional cosine transform of a discrete function $f(j)$ is described in Chapter 3, Section (3.2.2.6) (the cosine transform has a superior energy compaction property when compared with other transforms). Figure 3.13 (Section 3.3.2) shows a plot of the mean square error of an image with Markov process covariance as a function of block size for various transformations.[21] In this plot the 25% of the coefficients containing the largest variances were selected and the remainder discarded. It is seen from the figure that the cosine transform results in virtually the same energy compaction performance as the Karhunen-Loeve transform, known to be the best in the mean square error sense. The transform possesses a modified circular convolution-multiplication relationship which can be used in linear system analysis,[68], and can also be calculated using a simple fast computational algorithm whereas the conventional approach of implementing the discrete cosine transform involves the computation of a double fast Fourier transform (FFT) ($2N$ coefficients) employing complex arithmetic [41]. The alternative algorithm, known as the Fast Discrete Cosine Transform (FDCT), requires only real operations on a set of $N$ points and a general method for the derivation of the FDCT for any desired value of $N = 2^m$, $m > 2$ has also been found [44]. The method requires $(3N/2) \log_2 N - (3N/2) + 4$ real multiplications.
4.3.3 Statistical Properties of Cosine Transform Samples

The development of efficient quantization and coding methods for image transform coefficients requires an understanding of the statistical properties of the transform domain samples.

4.3.3.1 Statistical Mean and Variance

Suppose each sample of an original image, denoted by the function \( f(j,k) \) over spatial co-ordinates \((j,k)\) is considered as a two-dimensional stochastic process with spatial mean

\[
E\{f(j,k)\} \equiv f(j,k) = m
\]  

(4.5)

Then \( E\{f(j,k)\} \) can be replaced by \( m \) and the expected value of the two-dimensional cosine transform samples can be expressed as

\[
E(T(u,v)) = \frac{4mC(u)C(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \cos\left(\frac{(2j+1)u\pi}{2N}\right) \cos\left(\frac{(2k+1)v\pi}{2N}\right)
\]

(4.6)

or

\[
E(T(0,0)) = 2m
\]

(4.7)

and

\[
E(T(u,v)) = 0 \quad ; \quad (u,v) \neq 0
\]

where \( C(0) = \frac{1}{\sqrt{2}} \) and \( C(1), C(2), \ldots, C(N-1) = 1 \)
A knowledge of the variance of the transform coefficients is important in the design of quantizers for the transform coefficients, since the number of bits required to quantize any coefficient is normally set proportional to the logarithm of its variance or energy.

With the mean denoted as above the variance of transform samples can be computed from

\[ \sigma^2(0,0) = E \{ [T(0,0)]^2 \} - 4 m^2 \]  

and

\[ \sigma^2(u,v) = E \{ [T(u,v)]^2 \}; (u,v) \neq 0 \]  

4.3.3.2 Probability Density Model

Generally, the probability density function (pdf) of the transform coefficients is difficult to obtain because the pdf of the original image is not usually well defined. Nevertheless, a model of the pdf is necessary to facilitate the design of appropriate quantizers based on a minimum mean square, [69, 70], or minimum nth power, error [71] criterion. Since each cosine transform domain sample is formed from the cosine weighted sum of all the pixels in the original block, it has been suggested [63] that the central limit theorem [72] tells us that the probability density functions can be modelled approximately by Gaussian densities, i.e.
\[ P(u,v)(T) = \frac{1}{\sqrt{2\pi} \sigma(u,v)} \exp\left\{-\frac{(T - E[T(u,v)])^2}{2\sigma^2(u,v)}\right\} \] (4.10)

where \( E[T(0,0)] = 2m \)

and \( E[T(u,v)] = 0 \) for \( u,v \neq 0 \)

(see comments in Section 4.3.4.1, pp. 107 below).

**4.3.4 Quantization**

Quantization is the process of mapping a set of discrete input levels \( \{X\} \) into a set of discrete output levels \( \{Y\} \) according to a chosen characteristic. It takes place by assigning the value \( Y_i \) to \( X \) if \( X \) lies between the values \( d_i \) to \( d_{i+1} \) where \( d \) and \( Y \) are decision and reconstruction levels respectively as shown in Figure 4.3. A uniform quantizer is one in which all bin widths are equal. Non-uniform quantizers allow different bins to have different widths as shown in Figure 4.4(b). For nonuniform quantization, given the probability density function (pdf) of a signal, the decision and reconstruction levels of the quantizer that minimizes the mean square quantization error can be found \([69],[86, 87]\), and there are several quantizer designs available that offer various tradeoffs between simplicity and performance.

**4.3.4.1 The Optimum Mean Square or Lloyd-Max Quantizer**

Here the criterion is to minimize the root mean-square quantization error for a fixed number of quantization
Figure 4.3 Quantizer Decision and Reconstruction levels
FIG. 4.4 - QUANTIZATION CHARACTERISTICS

a) UNIFORM QUANTIZER

b) NON-UNIFORM QUANTIZER
levels. This error is defined as

\[
q_{\text{ms}} = \sum_{i=1}^{L} \left( \frac{d_{i+1} - d_i}{2} \right) \int_{d_i}^{d_{i+1}} (X - Y_i)^2 P(X) \, dX
\]  

(4.12)

where \( P(X) \) is the probability density function (pdf) of the transform coefficient (represented by discrete variable \( X \)), \( Y_i \) is the reconstructed value from the output of the quantizer and \( L \) is the number of quantization levels. The mean square error is minimised is the following two criteria are met:

\[
d_i = \frac{Y_i + Y_{i-1}}{2}
\]  

(4.13)

\[
\int_{d_i}^{d_{i+1}} P(X) (Y_i - X) \, dX = 0
\]  

(4.14)

Equation 4.13 and 4.14 describe the overall relationship for the optimum quantizer. Equation 4.13 states that the decision level \( d_i \) should lie half way between \( Y_i \) and \( Y_{i-1} \). Equation 4.14 shows \( Y_i \) (the reconstruction level or quantizer output) to be the centroid of the area under \( P(X) \) between \( d_i \) and \( d_{i+1} \).

The probability density function of transform coefficients has been modelled as a Rayleigh density for the DC coefficients and a Gaussian or Laplacian density for the ac coefficients. Ghanbari and Pearson[91] found that the Hadamard
ac coefficients approximate to a Gamma pdf and Chen and Smith modelled the pdf of the dc and ac coefficients as Rayleigh and Gaussian densities respectively. The logarithmic model for the variances of ac transform coefficients has also been used[89-90]. However, from a study of the histograms of the DCT coefficients (Figure 4.5) of the "Girl" picture (shown in Figure 4.6), it can be seen that the pdf of the ac coefficients approximates more to a Laplacian than a Gaussian distribution.

Table 4.I shows the decision and reconstructions levels of a Max-quantizer for the Laplacian distribution (L=2 to L=128). Thus, for example for a fixed number of quantization levels, for example, 128 (7 bits), if the input value lies between the decision levels 4.85 and 5.21, the reconstructed output value will be 5.01. If the input value is greater than 8.02 (for the same number of quantization levels L), the reconstructed value will be 8.738, and so on.

4.3.4.2 The Compander

The compander is a compression-expansion device which can be used to perform non-linear quantization of signals where a uniform quantizer is preceded and succeeded by non-linear transformations as shown in Figure (4.7)[87],[94].
FIG. 4.5 HISTOGRAMS OF THE DCT COEFFICIENTS OF THE "GIRL" PICTURE.
FIG. 4.5 HISTOGRAMS OF THE DCT COEFFICIENTS OF THE "GIRL" PICTURE.
FIG. 4.6 THE ORIGINAL "GIRL" PICTURE.
Table 4.1 Placement of decision and reconstruction levels for the Laplacian Quantizer

<table>
<thead>
<tr>
<th>Bits</th>
<th>$X_1$</th>
<th>$Y_1$</th>
<th>Bits</th>
<th>$X_1$</th>
<th>$Y_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00000</td>
<td>0.70711</td>
<td>0</td>
<td>0.00000</td>
<td>0.01639</td>
</tr>
<tr>
<td></td>
<td>1.12527</td>
<td>1.83397</td>
<td>0</td>
<td>0.03304</td>
<td>0.04967</td>
</tr>
<tr>
<td>2</td>
<td>0.00000</td>
<td>0.23340</td>
<td>0</td>
<td>0.06660</td>
<td>0.08351</td>
</tr>
<tr>
<td></td>
<td>0.53397</td>
<td>0.82946</td>
<td>0</td>
<td>0.10700</td>
<td>0.11788</td>
</tr>
<tr>
<td></td>
<td>1.25274</td>
<td>1.67251</td>
<td>0</td>
<td>0.13535</td>
<td>0.15292</td>
</tr>
<tr>
<td></td>
<td>1.09750</td>
<td>0.18936</td>
<td>0</td>
<td>0.17059</td>
<td>0.18936</td>
</tr>
<tr>
<td></td>
<td>2.37965</td>
<td>3.06860</td>
<td>0</td>
<td>0.20453</td>
<td>0.22451</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>0.12399</td>
<td>0</td>
<td>0.24290</td>
<td>0.26128</td>
</tr>
<tr>
<td></td>
<td>0.26442</td>
<td>0.40484</td>
<td>0</td>
<td>0.28000</td>
<td>0.29782</td>
</tr>
<tr>
<td></td>
<td>0.55675</td>
<td>0.72866</td>
<td>0</td>
<td>0.31779</td>
<td>0.33684</td>
</tr>
<tr>
<td></td>
<td>0.91984</td>
<td>1.11102</td>
<td>0</td>
<td>0.35625</td>
<td>0.37567</td>
</tr>
<tr>
<td></td>
<td>1.33443</td>
<td>1.57784</td>
<td>0</td>
<td>0.39474</td>
<td>0.41524</td>
</tr>
<tr>
<td></td>
<td>1.87764</td>
<td>2.17743</td>
<td>0</td>
<td>0.43323</td>
<td>0.45356</td>
</tr>
<tr>
<td></td>
<td>2.59722</td>
<td>3.01701</td>
<td>0</td>
<td>0.47161</td>
<td>0.49666</td>
</tr>
<tr>
<td></td>
<td>3.72421</td>
<td>4.43142</td>
<td>0</td>
<td>0.51762</td>
<td>0.53858</td>
</tr>
<tr>
<td>3</td>
<td>0.00000</td>
<td>0.06404</td>
<td>0</td>
<td>0.55996</td>
<td>0.58133</td>
</tr>
<tr>
<td></td>
<td>0.13220</td>
<td>0.20035</td>
<td>0</td>
<td>0.59315</td>
<td>0.62149</td>
</tr>
<tr>
<td></td>
<td>0.27320</td>
<td>0.34605</td>
<td>0</td>
<td>0.62726</td>
<td>0.65594</td>
</tr>
<tr>
<td></td>
<td>0.42248</td>
<td>0.49214</td>
<td>0</td>
<td>0.64000</td>
<td>0.66308</td>
</tr>
<tr>
<td></td>
<td>0.55497</td>
<td>0.63644</td>
<td>0</td>
<td>0.64134</td>
<td>0.66144</td>
</tr>
<tr>
<td></td>
<td>0.76322</td>
<td>0.85000</td>
<td>0</td>
<td>0.64134</td>
<td>0.66144</td>
</tr>
<tr>
<td></td>
<td>0.95548</td>
<td>1.05596</td>
<td>0</td>
<td>0.63357</td>
<td>0.65792</td>
</tr>
<tr>
<td></td>
<td>1.16697</td>
<td>1.27798</td>
<td>0</td>
<td>0.62282</td>
<td>0.64972</td>
</tr>
<tr>
<td></td>
<td>1.40199</td>
<td>1.52601</td>
<td>0</td>
<td>0.61271</td>
<td>0.63693</td>
</tr>
<tr>
<td></td>
<td>1.66448</td>
<td>1.80695</td>
<td>0</td>
<td>0.60261</td>
<td>0.62336</td>
</tr>
<tr>
<td></td>
<td>1.96891</td>
<td>2.10388</td>
<td>0</td>
<td>0.59233</td>
<td>0.60993</td>
</tr>
<tr>
<td></td>
<td>2.32214</td>
<td>2.41340</td>
<td>0</td>
<td>0.58206</td>
<td>0.59671</td>
</tr>
<tr>
<td></td>
<td>2.74464</td>
<td>2.78048</td>
<td>0</td>
<td>0.57179</td>
<td>0.58363</td>
</tr>
<tr>
<td></td>
<td>3.28050</td>
<td>3.58051</td>
<td>0</td>
<td>0.56153</td>
<td>0.57067</td>
</tr>
<tr>
<td></td>
<td>4.00074</td>
<td>4.42098</td>
<td>0</td>
<td>0.55127</td>
<td>0.55785</td>
</tr>
<tr>
<td></td>
<td>5.12949</td>
<td>5.83800</td>
<td>0</td>
<td>0.54098</td>
<td>0.54502</td>
</tr>
<tr>
<td>4</td>
<td>0.00000</td>
<td>0.03258</td>
<td>0</td>
<td>0.53061</td>
<td>0.55012</td>
</tr>
<tr>
<td></td>
<td>0.06617</td>
<td>0.09979</td>
<td>0</td>
<td>0.54081</td>
<td>0.56053</td>
</tr>
<tr>
<td></td>
<td>0.13449</td>
<td>0.16919</td>
<td>0</td>
<td>0.55101</td>
<td>0.57094</td>
</tr>
<tr>
<td></td>
<td>0.20507</td>
<td>0.24094</td>
<td>0</td>
<td>0.56121</td>
<td>0.58096</td>
</tr>
<tr>
<td></td>
<td>0.27604</td>
<td>0.31199</td>
<td>0</td>
<td>0.57141</td>
<td>0.59100</td>
</tr>
<tr>
<td></td>
<td>0.35566</td>
<td>0.39213</td>
<td>0</td>
<td>0.58161</td>
<td>0.61004</td>
</tr>
<tr>
<td></td>
<td>0.41305</td>
<td>0.47199</td>
<td>0</td>
<td>0.59182</td>
<td>0.63096</td>
</tr>
<tr>
<td></td>
<td>0.51348</td>
<td>0.55498</td>
<td>0</td>
<td>0.60202</td>
<td>0.65088</td>
</tr>
<tr>
<td></td>
<td>0.58916</td>
<td>0.64135</td>
<td>0</td>
<td>0.61223</td>
<td>0.67080</td>
</tr>
<tr>
<td></td>
<td>0.68837</td>
<td>0.73139</td>
<td>0</td>
<td>0.62244</td>
<td>0.69072</td>
</tr>
<tr>
<td></td>
<td>0.77841</td>
<td>0.82542</td>
<td>0</td>
<td>0.63266</td>
<td>0.71064</td>
</tr>
<tr>
<td></td>
<td>0.87434</td>
<td>0.92381</td>
<td>0</td>
<td>0.64288</td>
<td>0.73056</td>
</tr>
<tr>
<td></td>
<td>0.97541</td>
<td>1.02701</td>
<td>0</td>
<td>0.65310</td>
<td>0.75048</td>
</tr>
<tr>
<td></td>
<td>1.08125</td>
<td>1.13550</td>
<td>0</td>
<td>0.66332</td>
<td>0.77040</td>
</tr>
<tr>
<td></td>
<td>1.19267</td>
<td>1.24984</td>
<td>0</td>
<td>0.67354</td>
<td>0.79032</td>
</tr>
<tr>
<td></td>
<td>1.31026</td>
<td>1.37069</td>
<td>0</td>
<td>0.68376</td>
<td>0.81024</td>
</tr>
<tr>
<td></td>
<td>1.43478</td>
<td>1.49887</td>
<td>0</td>
<td>0.69398</td>
<td>0.83016</td>
</tr>
<tr>
<td></td>
<td>1.56709</td>
<td>1.63530</td>
<td>0</td>
<td>0.70420</td>
<td>0.85008</td>
</tr>
<tr>
<td></td>
<td>1.70829</td>
<td>1.78110</td>
<td>0</td>
<td>0.71442</td>
<td>0.87000</td>
</tr>
<tr>
<td></td>
<td>1.85930</td>
<td>1.92766</td>
<td>0</td>
<td>0.72464</td>
<td>0.89000</td>
</tr>
<tr>
<td></td>
<td>2.02216</td>
<td>2.06467</td>
<td>0</td>
<td>0.73486</td>
<td>0.91000</td>
</tr>
<tr>
<td></td>
<td>2.19851</td>
<td>2.20534</td>
<td>0</td>
<td>0.74508</td>
<td>0.93000</td>
</tr>
<tr>
<td></td>
<td>2.37929</td>
<td>2.49146</td>
<td>0</td>
<td>0.75530</td>
<td>0.95000</td>
</tr>
<tr>
<td></td>
<td>2.62565</td>
<td>2.71466</td>
<td>0</td>
<td>0.76552</td>
<td>0.97000</td>
</tr>
<tr>
<td></td>
<td>2.83377</td>
<td>2.96188</td>
<td>0</td>
<td>0.77574</td>
<td>0.99000</td>
</tr>
<tr>
<td></td>
<td>3.10245</td>
<td>3.24303</td>
<td>0</td>
<td>0.78596</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>3.40874</td>
<td>3.56725</td>
<td>0</td>
<td>0.79618</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>3.71217</td>
<td>3.94715</td>
<td>0</td>
<td>0.80640</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>4.14107</td>
<td>4.41820</td>
<td>0</td>
<td>0.81663</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>4.71124</td>
<td>5.01276</td>
<td>0</td>
<td>0.82686</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>5.43921</td>
<td>5.88106</td>
<td>0</td>
<td>0.83709</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>6.27528</td>
<td>7.03344</td>
<td>0</td>
<td>0.84732</td>
<td>0.99999</td>
</tr>
</tbody>
</table>
The input signal $X$ is first passed through a non-linear transformation $f(.)$ to yield $\hat{X}$. $\hat{X}$ is then uniformly quantized to give $Y$, which is non-linearly transformed by $g(.)$ to yield the output $Y$. The overall transformation from $X$ to $Y$ is that of a non-linear quantizer. The functions $f(.)$ and $g(.)$ are determined so that the overall system approximates the Max quantizer. Table 4.II [48] contains the functions $f(.)$ and $g(.)$ for the Gaussian, Rayleigh and Laplaceian probability densities.

4.3.5 Adaptive Cosine Transform Coding Schemes for Monochrome Images

The level of activity within an image block determines the a.c. energy within the transform domain, thus the greater the level of image detail within a block, the more a.c. energy there is and the higher the "activity". (The d.c. coefficient in the transform domain, however, only determines the average brightness level). After dividing the picture into sub-blocks, the latter are classified into various groups according to different strategies, and bit assignment is then carried out so as to achieve adaptivity within the system by assigning more bits to coefficients of higher energy and fewer bits to those of lower energy within the sub-blocks. Such a coding method provides good quality reconstruction for high activity regions and also achieves efficient coding for those of low activity. For adaptivity to be successful the range of adaptivity must be sufficiently large and coding in small sub-blocks
FIG. 4.7 - THE COMPANDER

TABLE 4.11 - COMPANDING QUANTIZATION TRANSFORMATION (AFTER REF. 48)

<table>
<thead>
<tr>
<th>Probability Density</th>
<th>Forward Transformation</th>
<th>Inverse Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>$p(f) = (2\pi \sigma^2)^{-1/2} \exp \left[-\frac{f^2}{2\sigma^2}\right]$</td>
<td>$g = \frac{1}{\sqrt{2\pi}} \exp \left[-\frac{y^2}{2}\right]$</td>
</tr>
<tr>
<td>Rayleigh</td>
<td>$p(f) = \frac{1}{\sigma^2} \exp \left[-\frac{f^2}{2\sigma^2}\right]$</td>
<td>$g = \frac{1}{\sigma} \exp \left[-\frac{y^2}{2\sigma^2}\right]$</td>
</tr>
<tr>
<td>Laplacian</td>
<td>$p(f) = \frac{\sigma}{2} \exp \left[-\frac{</td>
<td>f</td>
</tr>
<tr>
<td></td>
<td>$\sigma = \frac{\sigma}{\sqrt{2}}$</td>
<td>$g = \frac{1}{\sigma} \exp \left[-\frac{</td>
</tr>
</tbody>
</table>
meets this requirement and is also computationally efficient.

4.3.5.1 System Description

Figure 4.8 shows the block diagram of an adaptive cosine transform coding system for monochrome images. In the system to be described, the entire image is divided into a number of 16 x 16 pixel sub-blocks, and these sub-blocks are transformed using a two-dimensional discrete cosine transform (DCT). The conventional measure of the level of activity of any sub-block can be computed using the absolute values, or the sum of the logarithms of the coefficient values within that sub-block.

To reduce the system complexity and computational load, instead of calculation of the a.c. coefficient energy or magnitude within the sub-blocks using all 256 (16 x 16) samples, the level of activity can be calculated by summation of only the (64) lower order coefficients in an (8 x 8) zonal sub-block, and/or the (16) lower order coefficients in a (4 x 4) zonal sub-block. The transform blocks are then classified into four groups as before. The classification map is then constructed and the variance and standard deviation matrices for each class also calculated. From the variances, in the case of the four classes, four bit allocation matrices are computed. For the purpose of quantization the normalization factor and the normalization coefficients are determined. The normalization operation
FIG. 4.8 BLOCK DIAGRAM OF THE ADAPTIVE COSINE TRANSFORM CODING SYSTEM.

(a) ENCODER
FIG. 4.8 (CONT.)  b) DECODER
carried out results in normalized transform coefficients within each class which are then non-uniformly quantized and adaptively coded. The quantization strategy which has been used is Max's optimal quantization scheme illustrated in Section 4.3.4.1 for a Laplacian distribution of quantization threshold levels as shown in Table 4.I. The quantization output levels are coded in binary form and then passed to a variable input-fixed output rate buffer. At the receiver the fixed input-variable output rate buffer receives the binary data, and the output of the buffer is then decoded according to the received bit allocation matrix and inverse transformed before display. The buffering techniques used are discussed in Chapter V.

4.3.5.2 The Classifier

Figure 4.9 shows a typical classification process in which the transform sub-blocks are applied to the input of the classifier to be sorted by level of activity. The activity index in the present work is determined by the variance of the sub-block, or the weighted sum of the absolute values of the transform coefficients as shown in Figure 4.9.

4.3.5.2.1 Classification Based On a.c. Energy

The a.c. energy in the \((m,1)\)th sub-block of \((N\times N)\)th dimension can be defined as:

\[
E_{m,1} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} [T_{m,1}(u,v)]^2 - [T_{m,1}(0,0)]^2
\]

\[4.15\]
FIG. 4.9 A TYPICAL CLASSIFICATION PROCESS.
\[ N = 4, 8, \text{ or } 16 \]
\[ m, l = 1, 2, \ldots, M/16^\ast \]

and the variance of the \((m, l)\)th sub-block is defined as

\[ \sigma_{m, l}^2 = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} [T_{m, l}(u, v) - \mu_{m, l}]^2 \quad (4.16) \]

for \((u, v) \neq (0, 0)\)

where

\[ \mu_{m, l} = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T_{m, l}(u, v) \quad (4.17) \]

and \(u\) is the mean value of the a.c. coefficients, within the \((m, l)\)th block. Since the mean value of the a.c. coefficients is normally very small, equation 4.16 becomes

\[ \sigma_{m, l}^2 = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} [T_{m, l}(u, v)]^2 \quad (4.18) \]

For the discrete cosine transform (DCT) the property of energy compaction of the transform coefficients into the upper left hand corner of the array gives the possibility of reducing the number of calculations needed. Thus classification of the block (activity index) can depend on only the few high magnitude coefficients in the upper left corner

\* where the image has \(M \times M\) elements
of each block as shown in Figure 4.10 where the distribution of the log absolute values of transform coefficients within a sub-block is presented in three dimensions for the "Girl" picture using the DCT with a transform block size of 16 x 16. In this case Equation 4.18 becomes

$$A_I = \sigma^2_{m,1} = \frac{1}{R^2} \sum_{u=0}^{R-1} \sum_{v=0}^{R-1} [T_{m,1}(u,v)]^2$$

(4.19)

where $R^2 = N^2 - k$, and $k$ is the number of discarded transform coefficients, $N \times N$ is the dimension of the block and $R \times R$ is the dimension of the restricted region chosen to define the activity.

Figure 4.11 shows the typical cumulative probability distribution of a.c. energy within a set of sub-blocks. The energy has been classified into four levels nonuniformly separated by the 0.25, 0.50, and 0.75 points of the cumulative probability distribution, such that each class contains 1/4 of the total number of sub-blocks. Classification of the transform sub-blocks into equally populated levels provides a simple way of ensuring that the average coding rate over the entire image is maintained. The number of classification levels is generally dependent upon the image size, the relative population of activity levels, the degree of change of activity within the image, and the average number of bits required to code the compressed data. While not necessarily optimum, four equally populated levels of
FIG. 4.10 - LOGARITHM OF THE ABSOLUTE COEFFICIENT VALUE WITHIN A SUB-BLOCK OF SIZE (16 x 16) FOR THE "GIRL" PICTURE.
FIG. 4.11 CUMULATIVE PROBABILITY DISTRIBUTION OF A.C. ENERGY WITHIN SUB-BLOCKS.
classification are sufficient to code images at an average rate of 0.5 bits per element. Figure 4.12 shows the distribution of the normalized energy within the transformed image for a block size of (16 x 16). Figure 4.13 shows the distribution of the normalized energy estimate within the transformed image for (8 x 8) zonal sub-blocks and Figure 4.14 shows the results for R = 4.

From Figures 4.12 -14 it can be seen that the probability density of the activity of a particular image such as the "Girl" picture is approximately an exponentially decreasing function of energy. For many image types this relationship holds [63],[65], and this indicates that most sub-blocks are relatively quiet and only small numbers of sub-blocks have significant activity. Consequently, a strong motivation for adaptivity is available and therefore coding at relatively low rates is acceptable for a large fraction of image sub-blocks. A small additional bandwidth is required to accommodate the small number of high activity sub-blocks. In addition, the comparison between the energy estimate for (16 x 16) elements blocks and (8 x 8) zonal sub-blocks is shown in Figure 4.15. In this Figure the relation between the activity of the 16 x 16 blocks and the 8 x 8 zonal sub-blocks of the picture can be seen to be approximately linear. There is thus the possibility of using a small number of coefficients in the top left hand corner of the transformed block to classify the picture, and this measure is used in the following work to reduce the computational burden of processing three different
RELATIVE FREQUENCY

FIG. 4.12 16 x 16 BLOCK ENERGY DISTRIBUTION FOR THE ENTIRE IMAGE.
FIG. 4.13 - 8 x 8 ZONAL SUB-BLOCK ENERGY DISTRIBUTION FOR THE ENTIRE IMAGE.
FIG. 4.14 - 4 x 4 ZONAL SUB-BLOCK ENERGY DISTRIBUTION FOR THE ENTIRE IMAGE
FIG. 4.15 - COMPARISON BETWEEN THE ENERGY ESTIMATE FOR (16 x 16) ELEMENT BLOCKS AND (8 x 8) ELEMENT ZONAL SUB-BLOCKS OVER THE ENTIRE "GIRL" PICTURE, (i.e THE "ACTIVITY INDEX" OF (16 x 16) BLOCKS VERSUS THE "ACTIVITY INDEX" OF (8 x 8) ZONAL SUB-BLOCKS)
pictures, "Girl", "Testcard", and "Flat" shown in Figures 4.6 and 4.16(a) respectively. These pictures are band limited to 2.5 MHz, sampled at 5.5 MHz, composed of 256 lines with 256 elements per line, each quantized to one of 256 luminance levels between 0 and 255. The pdf ranges of "Girl", "Testcard" and "Flat" are 98-255, 46-180, and 0-198 respectively. Figure 4.16(b,c,d) shows the histogram of the luminance levels for these pictures.

After the classification of blocks into four categories, the classification maps indicate the appropriate bit allocation matrices and are part of the overhead information which will be needed by the receiver. Figure 4.17 shows the classification maps for the images of Figures 4.6 and 4.16(a).

4.3.5.2.2 Classification Of The Weighted Sum Of The Absolute Value Of The Transformed Coefficients

The classification of blocks using the weighted sum of the absolute values of the transformed coefficients has also been investigated for the purpose of reducing system complexity. The activity index "A" has been defined by \[55\].

\[
A_m,1 = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |T(u,v)| \quad \text{for } (u,v) \neq 0 \tag{4.20}
\]

where \(T(u,v)\) are the transform coefficients. The summation omits the d.c. coefficient because it is normally the only coefficient with non-zero mean. In this way, a subpicture with no detail would have \(A = 0\).
FIG. 4.16(a) THE ORIGINAL PICTURES
FIG. 4.16(b) LUMINANCE LEVEL HISTOGRAM FOR PROCESSED "GIRL" PICTURE
FIG. 4.16(c) LUMINANCE LEVEL HISTOGRAM FOR PROCESSED "TESTCARD" PICTURE
FIG. 4.16(d) LUMINANCE LEVEL HISTOGRAM FOR PROCESSED "FLAT" PICTURE
FIG. 4.17 CLASSIFICATION MAPS FOR GIRL, TESTCARD, AND FLAT IMAGES RESPECTIVELY (EACH PICTURE DIVIDED INTO (16 x 16) BLOCKS OF (16 x 16) ELEMENTS)
FIG. 4.17 (CONTINUED)
The system in Figure 4.8 has been used in this investigation with the exception that the above method of computation of the activity index based on the use of the classification in Figure 4.9 has been employed. Results will be found in Section 4.4.

4.3.5.3 Bit Assignment

In order to minimise the mean square error for a given number of bits allocated to transform coefficients, the optimum bit assignment makes the average quantization error of each transform coefficient sequence the same, and this requires that bits be assigned to the transform coefficient sequences in proportion to the logarithm of their variances. The bit allocation matrix for each class of image activity is determined by the computation of the ensemble average of variance of transform coefficients within each class. The variance of the transform coefficients of the \((u,v)\)th location can be computed by the equation:

\[
\sigma^2_{u,v} = \frac{1}{M} \sum_{k=1}^{M} [T_{u,v}(k) - \mu_{u,v}]^2
\]

(4.21)

where \(M = 64\) for an image of 256 square blocks, and \(u,v = 1, 2, \ldots, N\). \(N \times N\) is the block size, \(T_{u,v}\) is the amplitude of the transformed coefficient at location \((u,v)\) and

\[
\mu_{u,v} = \frac{1}{M} \sum_{k=1}^{M} T_{u,v}(k)
\]

(4.22)
Once the variances of the individual coefficients in each class within the sub-block have been computed, the bit allocation matrix for that class can be determined. The bit allocation $N_{bk}(u,v)$ has been adaptively assigned for each allocation matrix, and is based upon a relationship from rate distortion theory [72-73,131], where it is stated that the output of a source can be transmitted with average distortion $D$ if the transmission rate is at least $R(D)$, where $R(D)$ is the rate-distortion function. For a source with Gaussian probability density function and mean square distortion measure $D$, the relation between $R(D)$ and $D$ is given by Shannon [132].

$$R(D) = N_{bk}(u,v) = \frac{1}{2} \log_2 \frac{\sigma_k^2(u,v)}{D}, 0 \leq D \leq \sigma_k^2(u,v)$$

$$= 0, \quad D > \sigma_k^2(u,v)$$

for $(u,v) \neq (0,0)$ \hspace{1cm} (4.23)

where $\sigma_k^2(u,v)$ is the variance of the individual coefficient as defined in Equation 4.21, and $N_{bk}(u,v)$ is the number of bits per element needed for transmission. The parameter $D$ may differ from class to class, and bits are assigned by initializing $D$ and iteratively calculating the summation of $N_{bk}(u,v)$ in Equation 4.23 until the desired total number is achieved. For a given transmission bit rate and picture quality, Figure 4.18 shows examples of the values of $D$ and the total number of bits in each class for the "Girl" picture, and Figure 4.19 shows bit allocation matrices for
FIG. 4.18  D-VERSUS-B FOR THE "GIRL" PICTURE WHERE D IS THE DISTORTION PARAMETER AND B IS THE TOTAL NUMBER OF BITS WITHIN EACH CLASS (CLASS ONE HAS THE HIGHEST ACTIVITY)
"Girl", "Testcard" and "Flat", respectively.

The third step in classification is to compute the standard deviation matrices $\sigma_k(u,v)$ from which the normalization factor $C$ can be determined. Referring to the block diagram of Figure 4.8 the incoming transform coefficient must be normalized before the quantization operation is carried out, to yield transform coefficients with unit variance, and to exclude any clipping effects due to the quantization operation. The normalization coefficient matrices are given by:

$$\sigma_k'(u,v) = C \cdot \frac{N_{bk}(u,v) - 1}{2} (4.24)$$

where $N_{bk}$ is the number of bits needed for each coefficient and is defined in Equation 4.23. In Equation 4.24 if the number of bits $N_{bk}(u,v)$ is one, the normalization factor $C$ will equal $\sigma_k'(u,v)$. Since $\sigma_k'(u,v)$ is simply the estimation of $\sigma_k(u,v)$, $C$ can be determined during the bit assignment process by setting it equal to the maximum standard deviation of those elements in the variance matrix which were assigned one bit. The maximum value is selected rather than the average to avoid excessive clipping, and eight bits are assigned to the d.c. coefficients to avoid brightness differences at sub-block boundaries. The normalization factor $C$ must of course be transmitted to the receiver. A normalization factor assigned to each transform sample might
FIG. 4.19(a) 16 x 16 BIT ALLOCATION MATRICES FOR "GIRL" PICTURE
WITH AVERAGE BIT RATE EQUAL TO 0.96
FIG. 4.19(b) 16 x 16 BIT ALLOCATION MATRICES FOR "TESTCARD" PICTURE WITH AVERAGE BIT RATE EQUAL TO 0.92
FIG. 4.19(c) 16 x 16 BIT ALLOCATION MATRICES FOR "FLAT" PICTURE
WITH AVERAGE BIT RATE EQUAL TO 1.56
yield slightly better performance but would lead to excessive overhead requirement. From Equation 4.24 and for the transform coefficients $T_k(u,v)$ in class $K$, the normalized coefficients are given by:

$$T_{k,N}(u,v) = \frac{T_k(u,v)}{\sigma_k(u,v)} \quad (4.25)$$

where the $T_{k,N}(u,v)$ form the input to the quantizer.

For a given number of bits $N_{bk}(u,v)$, the number of quantization levels $L$ is given by

$$L = 2^{N_{bk}(u,v)} \quad (4.26)$$

and the optimum placement of levels within the coefficient amplitude range to provide minimum distortion is given by the Max quantizer described in section 4.3.4.1.

The normalized coefficients $T_{k,N}(u,v)$ are thus optimally quantized with the number of quantization levels (bits) set according to the appropriate bit allocation matrices for the activity classes. Max's quantization scheme is used here since it gives the optimal result for a given probability density function, and the probability density model used in finding decision and reconstruction levels, (Table 4.I), is Laplacian, as defined in Equation (3.44 Section 3.4). The quantized coefficients are then coded into a variable word length binary data stream. This data and the necessary overhead information is then, after buffering, transmitted over the channel.
4.3.5.4 Overhead Information

The overhead information is necessary to decode the compressed image data at the receiving end of the channel. In the adaptive transform coding system which has been described, the total overhead information consists of:

- one classification map
- one or four normalization factors
- four bit allocation matrices.

This can be written as:

$$B = B_{c.m} + B_{n.f} + B_{b.a}$$  \hspace{1cm} (4.27)

where $B_{c.m}$, $B_{n.f}$, and $B_{b.a}$ are the average overhead code bits required to code the classification map, the normalization factor, and the bit allocation matrices respectively. They are represented by the following expressions:

$$B_{c.m} = \frac{2 \left( \frac{M}{16} \right)^2}{M^2} = \frac{1}{128} \text{ bits}$$  \hspace{1cm} (4.28)

$$B_{n.f} = \frac{C'}{M^2} \text{ bits}$$  \hspace{1cm} (4.29)

$$B_{b.a} = \frac{h(16)^2 \cdot 4}{M^2} \text{ bits}$$  \hspace{1cm} (4.30)

where $C'$ is the number of bits used to represent the normalization factor $C$. The factor $h$ is a function of the
compression ratio, which is defined in the transform domain as:

\[ CCR = \frac{M^2}{NTC} \]

where CCR is the coefficient compression ratio, \( M^2 \) is the total number of coefficients, and the NTC is the number of transmitted coefficients.

Utilizing an efficient scheme such as the Huffman code, \( h \) is equal to 1.62 for the 1 bit case and 1.41 for 0.5 bit case\(^{[63]}\), based upon the fact that the majority of bit assignments are equal to zero for large compression ratios. The overhead part of the data, of course, requires error protection to avoid catastrophic errors and produce reliable data transmission. It can be calculated that, for the errorless case, less than 0.034 bits of overhead information is needed to code a one bit monochrome image of 256 x 256 pixels. For a one bit monochrome image of 1024 x 1024 pixels, the overhead information is less than 0.01 bits.

4.4 EXPERIMENTAL RESULTS

This chapter has been concerned with techniques for transform image coding at low bit rates. Three techniques have been investigated and partially computer simulated. The first is coding using a threshold sampling method, the second uses zonal sampling, and the third is adaptive transform image coding using an "activity index" for the transformed picture blocks. The results obtained for these three systems are outlined below.
4.4.1 Threshold Sampling

In this system the head and shoulder "Girl" picture has been processed by division of the 256 x 256 picture elements into (16 x 16) blocks each of which has 256 elements. The transformation of each block is then carried out using the two dimensional discrete cosine transform (DCT). To determine the value of a specific threshold the activities of different blocks from different regions of the image are measured (selection of any desired transformed block can be made). The mean square error criterion is used to estimate the effect of a range of thresholds on individual blocks. Table 4.III illustrates this operation where the threshold values are set to value in the range 0.1 to 100. The number of truncated coefficients, the values of threshold, the normalized mean square errors, and the number of operations are computed for a particular block. Having this information for low and high detailed regions of the image, suitable overall values of threshold can be determined. Tables 4.III and 4.IV illustrate the effect of the same truncation within low and high detail regions respectively. Having determined the threshold level, coefficients which have values above the threshold are retained. The inverse discrete cosine transform is then applied to reconstruct the original image. The performance of this system is shown in Figure 4.20 for various threshold levels corresponding to transmitting an average within each block of only 25.6, 11.5 and 4.6 coefficients respectively, out of a total of 256
Table 4.III. Effect of threshold levels on coding of low detail region in "Girl" picture (block number 1x15)

<table>
<thead>
<tr>
<th>SN</th>
<th>TH</th>
<th>NMSE %</th>
<th>NTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.10</td>
<td>0.000</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>0.80</td>
<td>0.000</td>
<td>53</td>
</tr>
<tr>
<td>3</td>
<td>2.70</td>
<td>0.003</td>
<td>150</td>
</tr>
<tr>
<td>4</td>
<td>6.40</td>
<td>0.014</td>
<td>234</td>
</tr>
<tr>
<td>5</td>
<td>12.5</td>
<td>0.023</td>
<td>250</td>
</tr>
<tr>
<td>6</td>
<td>21.60</td>
<td>0.025</td>
<td>251</td>
</tr>
<tr>
<td>7</td>
<td>34.30</td>
<td>0.042</td>
<td>254</td>
</tr>
<tr>
<td>8</td>
<td>51.20</td>
<td>0.042</td>
<td>254</td>
</tr>
<tr>
<td>9</td>
<td>72.90</td>
<td>0.042</td>
<td>254</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>0.085</td>
<td>255</td>
</tr>
</tbody>
</table>

SN - Sequence number
TH - Threshold level
NMSE % - Normalized mean square error (percentage)
NTC - Number of truncated coefficients
Table 4.IV. Effect of threshold levels on coding of high detail region in "Girl" picture (Block number 8x10)

<table>
<thead>
<tr>
<th>SN</th>
<th>TH</th>
<th>NMSE %</th>
<th>NTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.10</td>
<td>0.000</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>0.8</td>
<td>0.000</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>2.70</td>
<td>0.002</td>
<td>92</td>
</tr>
<tr>
<td>4</td>
<td>6.40</td>
<td>0.014</td>
<td>139</td>
</tr>
<tr>
<td>5</td>
<td>12.60</td>
<td>0.062</td>
<td>182</td>
</tr>
<tr>
<td>6</td>
<td>21.60</td>
<td>0.179</td>
<td>217</td>
</tr>
<tr>
<td>7</td>
<td>34.30</td>
<td>0.446</td>
<td>241</td>
</tr>
<tr>
<td>8</td>
<td>51.20</td>
<td>0.538</td>
<td>245</td>
</tr>
<tr>
<td>9</td>
<td>72.90</td>
<td>0.628</td>
<td>247</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1.09</td>
<td>252</td>
</tr>
</tbody>
</table>

**SN** - Sequence number  
**TH** - Threshold level  
**NMSE %** - Percentage normalized mean square error  
**NTC** - Number of truncated coefficients
FIG. 4.20 RESULTS OF THRESHOLD SAMPLING SYSTEM
(a) THE ORIGINAL
(b) NMSE% = 0.02 WITH THRESHOLD LEVEL = 10
(c) NMSE% = 0.09 WITH THRESHOLD LEVEL = 20
(d) NMSE% = 0.25 WITH THRESHOLD LEVEL = 50
coefficients in each case. For example, the degradation in Figure 4.20(c) is unnoticeable, whilst the number of transform coefficients has been substantially reduced.

4.4.2 Zonal Sampling

Fixed zonal coding has also been investigated. Two schemes have been examined using a rectangular zone. For a transform block size of (16 x 16), an (8 x 8) zone is taken for transmission and those coefficients outside this zone have been discarded in the first scheme. In the second zone of (4 x 4) coefficients is chosen. The results of both schemes, as applied to the "Girl" image shown in Figure 4.21. It can be seen that for the first scheme using an (8 x 8) zonal sub-block degradation in the picture is unnoticeable, but in the second some blocking effects in high detail regions do appear.

Fixed zonal coding techniques do not take into consideration the variation of activity within the transformed blocks of the image, however, and an adaptive technique is required to achieve high data compression rates. These two schemes are convenient, however, for the preliminary investigation of the computational reduction discussed in section 4.3.5.2.

4.4.3 Activity Index Classification Via A Simplified Energy Estimation Method

Two different approaches to adaptive transform coding related to activity index assignment have been investigated.
FIG. 4.21 RESULTS OF ZONAL CODING

ORIGINAL

(8 x 8) ZONAL SUB-BLOCK WITH NMSE = 0.08%

(4 x 4) ZONAL SUB-BLOCK WITH NMSE = 0.28%
One approach is to divide the transform blocks into a finite number of categories or classes, according to their activity index [55-59], [63], [65], [95-97], which represents the amount of "activity" or "detail" in each block. Bits are then distributed among classes according to their level of activity with more bits assigned to high activity class and vice versa.

In previous work [55-59], [63], [65], [95-97] implementing this approach, the a.c. energy of each whole block has been used as an activity index for class assignment. In this work an alternative approach to activity measurement has been considered, in order to reduce the amount of computation required and so simplify the system.

Actually, the most demanding operation from the computational point of view is the calculation of coefficient variances (Equations 4.16 and 4.18) which is carried out over all the coefficients of each block. Here a new classifier has been simulated (Section 4.3.5.2) to reduce the number of calculations.

In the classifier, instead of using all these coefficients in determining the variance or the summation of the absolute values of the coefficients to classify the picture blocks, only a zonal sub-block of 8 x 8 or 4 x 4 coefficients has been used, where the block contains those coefficients of lower order in both dimensions (i.e. in Equation 4.19 \( R = 4, 8 \)). From a knowledge of the activity index of each
block the reordering operation can be carried out (Figure 4.9). For i classes of activity, \(i-1\) activity thresholds are needed. Alternatively the reordered activity range can be divided into equally populated classes. Both methods have been used, and it has been found that the latter is more simple from the computational point of view for exact determination of the number of blocks within each class. By increasing the number of classes the performance of the system should be improved. However, the amount of overhead information will increase also.

In fact, increasing the number of classes from 2 to 8 shows a slight improvement (less than 0.5 dB in terms of SNR [97]) and therefore the processed images have here been classified using four classes only.

The performance of the adaptive transform coding system (Figure 4.8) with activity index based on the computation of the variance or the magnitude of the block coefficients is shown in Figures 4.22-4.30. In Figures 4.22-24 the performance of the system is shown for the conventional method of activity assignment using (16 x 16) element blocks for the "Girl", "Testcard" and "Flat" pictures respectively. Figures 4.25-27 show the system performance with the simplified activity assignment using (8 x 8) element zonal sub-blocks, and Figure 4.28 represents the results for (4 x 4) element zonal sub-blocks. The activity index assignment in Figures 4.22-28 has been based on the computation of the a.c. energy within each block or zonal sub-block. From Figures
FIG. 4.22 RESULT OF ADAPTIVE TRANSFORM IMAGE CODING USING THE CONVENTIONAL METHOD OF a.c ENERGY CLASSIFICATION WITH (16 x 16) ELEMENT BLOCKS.
(a) ORIGINAL "GIRL" PICTURE
(b) PROCESSED PICTURE
FIG. 4.23 RESULT OF ADAPTIVE TRANSFORM IMAGE CODING USING THE
CONVENTIONAL METHOD OF a.c ENERGY CLASSIFICATION WITH
(16 x 16) ELEMENT BLOCKS
(a) ORIGINAL "TESTCARD" PICTURE
(b) PROCESSED PICTURE
FIG. 4.24 RESULT OF ADAPTIVE TRANSFORM IMAGE CODING USING THE CONVENTIONAL METHOD OF a.c ENERGY CLASSIFICATION WITH (16 x 16) ELEMENT BLOCKS
(a) ORIGINAL "FLAT" PICTURE
(b) PROCESSED PICTURE
**FIG. 4.25** RESULT OF ADAPTIVE TRANSFORM IMAGE CODING SYSTEM USING THE ALTERNATIVE METHOD OF a.c ENERGY CLASSIFICATION WITH (8 x 8) ELEMENT ZONAL SUB-BLOCKS

(a) ORIGINAL "GIRL" PICTURE
(b) PROCESSED PICTURE
FIG. 4.26 RESULT OF ADAPTIVE TRANSFORM IMAGE CODING SYSTEM USING
THE ALTERNATIVE METHOD OF a.c ENERGY CLASSIFICATION
WITH (8 x 8) ELEMENT ZONAL SUB-BLOCKS
(a) ORIGINAL "TESTCARD" PICTURE
(b) PROCESSED PICTURE
FIG. 4.27 RESULT OF ADAPTIVE TRANSFORM IMAGE CODING USING THE ALTERNATIVE METHOD OF A.C ENERGY CLASSIFICATION WITH (8 x 8) ELEMENT ZONAL SUB-BLOCKS

(a) ORIGINAL "FLAT" PICTURE

(b) PROCESSED PICTURE
FIG. 4.28 RESULTS OF ADAPTIVE TRANSFORM IMAGE CODING SYSTEM USING THE ALTERNATIVE METHOD OF a.c ENERGY CLASSIFICATION WITH (4 x 4) ELEMENT ZONAL SUB-BLOCKS.
(a) PROCESSED "GIRL" PICTURE
(b) PROCESSED "TESTCARD" PICTURE
(c) PROCESSED "FLAT" PICTURE
4.22-28 it is clear that the degradation in the pictures is insignificant, whilst a reduction in computation of about 75% and 93% for the $8 \times 8$ and $4 \times 4$ cases, respectively, has been achieved.

Tables 4.V and 4.VI contain the parameters of an adaptive transform image coding system using $(8 \times 8)$ zonal sub-block ac energy classification resulting in the images of Figures 4.25-27. From the tables, with increasing distortion parameter "D" the number of bits required for each class, the total number of bits, and the average bit rate will all decrease, while the normalized mean square error will be increased. By changing "D" the performance of the system can be modified as desired.

In another approach, the magnitude of the DCT coefficients, instead of their squared value, has been used in Equation 4.19 as in Equation 4.20. This of course means that there are no multiplication operations at all in the evaluation of the activity index. Again, the number of computations in Equation 4.20 has been reduced by taking sub-blocks of $(8 \times 8)$ and $(4 \times 4)$. The performance when the sum of the magnitudes are evaluated by Equation 4.20 for $N = 8$ and $4$ is shown in Figures 4.29-30. It is clear that the performance is quite satisfactory whilst the amount of computation has been significantly reduced.
Table 4.V. Parameters of a (16 x 16) adaptive transform coding system classified by a.c. energy in (8 x 8) zonal sub-blocks.

<table>
<thead>
<tr>
<th>Source data</th>
<th>Class No.</th>
<th>Value of &quot;D&quot;</th>
<th>Number of bits in class</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIRL</td>
<td>1</td>
<td>5.5</td>
<td>367</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>342</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>68</td>
</tr>
<tr>
<td>TESTCARD</td>
<td>1</td>
<td>4.5</td>
<td>367</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>294</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>73</td>
</tr>
<tr>
<td>FLAT</td>
<td>1</td>
<td>11</td>
<td>593</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>7</td>
<td>566</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>363</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>6</td>
<td>80</td>
</tr>
</tbody>
</table>
Table 4.VI. Results for Table 4.V

<table>
<thead>
<tr>
<th>Source data</th>
<th>Average bit rate</th>
<th>NMSE %</th>
<th>Original Data Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIRL</td>
<td>0.96</td>
<td>0.13</td>
<td>98 - 255</td>
</tr>
<tr>
<td>TESTCARD</td>
<td>0.93</td>
<td>0.29</td>
<td>46 - 180</td>
</tr>
<tr>
<td>FLAT</td>
<td>1.56</td>
<td>0.57</td>
<td>0 - 198</td>
</tr>
</tbody>
</table>
FIG. 4.29 RESULT OF ADAPTIVE TRANSFORM IMAGE CODING SYSTEM USING THE ALTERNATIVE METHOD OF $a.c$ MAGNITUDE CLASSIFICATION WITH $(8 \times 8)$ ELEMENTS ZONAL SUB-BLOCKS.

(a) PROCESSED "GIRL" PICTURE  
(b) PROCESSED "TESTCARD" PICTURE  
(c) PROCESSED "FLAT" PICTURE
FIG. 4.30 RESULTS OF ADAPTIVE TRANSFORM IMAGE CODING SYSTEM USING THE ALTERNATIVE METHOD OF a.c MAGNITUDE CLASSIFICATION WITH (4 x 4) ELEMENT ZONAL SUB-BLOCKS
(a) PROCESSED "GIRL" PICTURE
(b) PROCESSED "TESTCARD" PICTURE
(c) PROCESSED "FLAT" PICTURE
4.5 DISCUSSION

From results obtained from implementing the three different systems, namely, threshold sampling, zonal sampling, and adaptive transform coding, it is clear that each performs differently. Threshold sampling performs better than zonal sampling, since the former preferentially selects high magnitude coefficients. A comparison between threshold sampling (Figure 4.20(c) - transmitting an average of 11.5 coefficients per block), and zonal sampling (Figure 4.21 - transmitting 4 x 4 zonal sub-blocks derived from each original 16 x 16 block) makes it clear that the performance of threshold sampling is better than that of zonal sampling in terms of mean square error reduction. However, threshold coding requires a large amount of address information to be transmitted to the receiver. In adaptive transform coding using activity index classification, the most difficult picture to classify and code is the "Flat" picture, mainly because it contains many horizontal and vertical edges and therefore more bits are required (an average of 1.56 bits per element), to achieve acceptable picture quality.

The "Girl" image is the easiest of the three images to classify and code, having the highest data compression figure with lowest NMSE as can be seen from Table 4.VI and Figure 4.25 in comparison with Figure 4.27. This is mainly due to the smoothness of the "Girl" picture, i.e. it does not contain many high frequency components, which
makes the classification method as well as the adaptation algorithm more efficient. As mentioned previously, adaptivity implies the allocation of more bits to active blocks (regions of sharp luminance transitions), and fewer bits to less active blocks (background areas). In consequence, the background of the pictures contains a small amount of noise due to the fact that the quantizers possess relatively few levels. On the other hand, the adaptive system encodes all the edges (which are the most important parts of the picture) accurately, and this justifies its use. In addition, increasing the average number of bits allotted to individual coefficients will increase the quality both visually and in terms of NMSE, since this allows a larger number of quantization levels to be used.

4.6 CONCLUSIONS

Any efficient coding scheme must take into consideration the picture quality required at the receiver and the complexity of the processing algorithm, since the latter is a major factor in any economical implementation of a digital video system. Thus, it is not so much a question of whether digital encoding of a video signal should be adopted for a digital transmission facility, but rather how sophisticated the coder will be from the point of view of complexity, which is a consequence of the algorithm used. Therefore, the simplifying of coding system complexity is one of the
important tasks in image processing. Two approaches to activity index assignment for simplifying classification operations have been investigated. From the results obtained for both methods it is found that the techniques are useful for reducing system complexity, whilst the picture degradations are insignificant as it can be seen from the figures presented. However, adaptive systems result in a variable data rate and therefore memory storage buffers are needed at the transmitter and receiver. This subject is considered in the next chapter.
CHAPTER V

BUFFERS FOR VARIABLE BIT RATE SYSTEMS
5.1 INTRODUCTION

In this chapter buffer storage is studied, since the design of the buffer is an important consideration in many aspects of data communications [101]. In the computer field messages entering the data-processing centre are read into buffers, which are main storage areas used for handling, queueing, and transferring message segments between all lines and queueing media and between queueing media and the application program area. One of the reasons for having buffers is to take advantage of the speed difference between line transmission and computer processing so that many users can have access to the data centre concurrently. Chang [102] has made a stochastic analysis of the buffer behaviour and procedures for calculating the required buffer size to satisfy a pre-specified probability of overflow within teleprocessing systems. The basic idea is that message arrivals at terminals are independent, but not necessarily exponentially distributed (as usually taken to be the case in such studies), and the usual assumption of simple streams or Poisson arrivals is not required. The asymptotic behaviour of the system was therefore studied by applying renewal theory to avoid many analytical difficulties. Renewal theory began as the study of particular problems in probability connected with the failure and replacement of components and it discusses the instantaneous replacement of components.
The theory of renewal processes in continuous time is dealt with by Cox [103]. In [102] the buffer size depends on a number of factors such as traffic rate, methods of queueing, message length and, processing speed. Other researchers have also contributed to this field. Birdsall et. al. [104], and later Dor [105] have analysed buffer behaviour with poisson input arrivals and constant output rate. Chu [106] has studied the behaviour of similar buffer models with multiple synchronous constant output rates, and for batch poisson arrivals and a single constant output rate [107]. In many data communication systems, input traffic is a mixture of bursts (strings of characters) and single characters. For example, video terminal outputs are in bursts and the tele-typewriter outputs are in characters [108-109]. Buffer behaviour with such mixed input traffic is studied by Chu [108]. The buffer overflow probability has been defined by Chu and is given by:

$$P_{of} = \frac{\text{offered load} - \text{carried load}}{\text{offered load}} = 1 - \frac{\beta}{\gamma} \quad (5.1)$$

where $\beta$ is the average character departure rate from the buffer (carried load), which is less than the average character arrival rate at the buffer (offered load) $\gamma$ is equal to $(\lambda_P + \lambda_C) \bar{L}$, where $\lambda_P$ is the arrival rate of single character input $X$ which is assumed to be poisson distributed, $\lambda_C$ is the arrival rate of the number of bursts $Z$ during a unit service interval (assumed to be poisson distributed), and $\bar{L}$ is the mean of the message length $L$ of the burst $Y$ which is assumed to be geometrically distributed. Another field
of buffer application is data compression systems (Figure 5.1). Such techniques have been used in many areas of communications such as voice, video, and telemetry transmission. Terms such as entropy reducing, information preserving, redundancy reduction, adaptive sampling, encoding, signal reduction and adaptive coding have been used to classify them. Andrews and Schwarz investigate and analyse buffer design for adaptive data compression techniques. The queueing analysis for buffer design is based upon the binomial distribution rather than the more usual poisson distribution model. To prevent either overflow or underflow of the buffer in their system a feedback control path is provided from the buffer to the compressor, to adjust the average output rate of the compressor to match the transmission rate. The same idea of buffer feedback techniques was used by Tescher and Cox figure 5.2(a,b). They define the "relative buffer status" to serve as a feedback condition from the output buffer to the compressor to induce a change in the distortion parameter for the purpose of maintaining the required average rate. In another approach Habibi has used feedback to adjust the accuracy of the analog to digital converter in an adaptive hybrid image coding scheme where the status of the buffer is checked every \( \Delta \) seconds (the time needed to process a sample block of a predetermined size), and the accuracy of the A/D converter is adjusted accordingly. This approach assumes that the bit rate averaged over \( \Delta \) second \( (\Delta = T_{k+1} - T_k) \) is related to
FIG. 5.1 DATA COMPRESSION CLASSIFICATION.
\( \bar{R} \), CHANNEL OPERATION RATE

\( R(t) \), REQUIRED CURRENT RATE

\( C(t) \), STOP (OR) CURVE AT TIME \( t \) (SEE REF. 112)

\( D(t) \), ESTIMATE OF \( D \) AT TIME \( t \)

\( N \), BUFFER SIZE IN BITS

\( \Delta \), CYCLE TIME

\( n \), NUMBER OF BITS TRANSMITTED DURING \( \Delta \)

\( B(t) \), BUFFER STATUS:
NUMBER OF BITS WITHIN BUFFER NORMALIZED BY THE BUFFER SIZE (AT NOMINAL RATE \( B = 1/2 \))

---

**FIG. 5.2**

(a) DEFINITION OF THE VARIOUS PARAMETERS ASSOCIATED WITH BUFFER FEEDBACK

(b) LINEARIZED BUFFER FEEDBACK.
the grey level resolution of the A/D converter over the same $\Delta$ seconds. The status of the buffer for every $M \times N_1$ data block is observed. The approximation to the grey level resolution of the A/D converter for the incoming block $b(k+1)$ is defined using the Newton-Raphson method of successive approximation and is given by:

$$b(k+1) = b(k) + \frac{1}{S(k)}(R(k+1) - R(k))$$  \hspace{1cm} (5.2)

where $k$ is the number of processed block, $R(k)$ is the number of binary digits generated for block number $k$, and $S(k)$ relates to the slope of $b(k)$ and is given by:

$$S(k) = \frac{R(k)}{b(k)}$$  \hspace{1cm} (5.3)

The value of $b(1)$ has been taken to be arbitrary resulting in a rate $R(1)$ (bits per block) for the first block. The process of successive approximation continues by choosing the value for $R(k+1)$, and the corresponding value for $b(k+1)$ is then obtained. In [112] Tescher has suggested estimating $R(k+1)$, which relates the problem to a coder where the data produced at variable rate $R(k)$ is buffered to a fixed rate system with $m$ bits/block. Habibi followed similar procedures to that of Tescher in which the buffer status is updated after coding each block as:

$$B(k) = \frac{R(k) - m}{M \times N} + B(k-1)$$  \hspace{1cm} (5.4)

where $B(k)$ is the updated buffer status, $m$ is the fixed rate (bits per block) and $M \times N$ is the block size. Then, depending upon $B(k)$,
the new rate is chosen as:

\[ R(k+1) = 1.5m - B(k) \]  \hspace{1cm} (5.5)

In the same area a transient mode buffer store for non-uniformly coded television (TV) has been investigated by Budrikis et al. [114]. The DPCM system uses a comma code, which is fed to the transmitter buffer to smooth the digit flow so that the transmission rate approximates the average rate of generation.

The sizes of both transmitter and receiver buffers are related to the bit transmission rate and storage delay. The transmitter buffer size is defined as:

\[ B_{Tx} = [R.D] - 1 \]  \hspace{1cm} (5.6)

and the receiver buffer size

\[ B_{Rx} = R.D \]  \hspace{1cm} (5.7)

where \( R \) is the nominal transmission rate (bits/sampling interval) and \( D \) is the nominal delay from encoding at transmitter to decoding at receiver. The random behaviour of the transmitter buffer is modelled as a first-order Markov chain, which is a stochastic process in which the future state of the system depends upon the present state but not on the past history of the system [115]. The average number of steps before transmitter buffer overflow is defined by:

\[ \bar{n}_1 = \frac{(B_{Tx} - i)}{(1 - \rho)L}, \quad \rho < 1 \]  \hspace{1cm} (5.8)
where \( \bar{n} \) represents the average number of sampling intervals from \( t = 0^+ \) to overflow.

\[
i \quad \text{is the number of locations occupied at } t = 0 \\
\text{(state of initial occupancy in the buffer)}
\]

\( \rho \) is the 'normalized' transmission rate \( (= R/L) \)

\( L \) is the average word length.

\( B_{Tx} \) is the transmitter buffer size.

In Budrikis' investigation when buffer overflow occurs the input to the buffer must be stopped until the buffer becomes completely empty, and this either results in data loss or an overflow channel must be provided, as suggested by the author.

The buffer with Huffman coded input within a differential pulse code modulation (DPCM) system has been investigated by Goyal and O'Neal [116] on the basis of average code length of data to test the entropy code and to determine buffer requirements for the system. They have determined the buffer size required for moderately active images. The buffer sizes required were 19000 and 13700 with average code lengths of (2.94 and 2.92) bits respectively, and with a transmission rate of 3.0 bits/sample to prevent buffer overflow. For other source data used an infinite buffer length was required, and they therefore reduced the number of quantization levels from 16 to 6 to prevent buffer overflow whenever the buffer becomes full.
Other contributors are Jelinek [117], and Jelinek and Schneider [118-119] in which the relationship between the probability of buffer overflow and efficient source encoding has been studied. On the other hand the importance of delay rather than storage has been pointed out by Gallager [120] who applied rate distortion theory to the coding problem with delay as the distortion measure. Again Humblet [121] has considered the problem of encoding randomly arriving messages so as to minimise delay. Hayes et.al [122a] studied the relationship between coding efficiency and delay, and the result of his study is that as the source becomes less bursty, delay grows without bound. Roskind [122b] has determined a strategy in which the delay is proportional to

$$\log \left[ \log \left( \frac{1}{1 - P} \right) \right]$$

as $P \to 1$  \hspace{1cm} (5.9)

where $P$ is the probability of the source producing a data digit, and the system produces either a blank or a data digit once per second. Recently a Joint European Project (involving, amongst others, the U.K., France, Germany and Italy) has been described by Carr, Temime, Clapp and Jolivet in which the problem of buffering overflow and underflow in a practical conditional replenishment video system is considered [123]. The investigation is carried out on the basis that areas in the moving image are predictively coded using differential pulse code modulation (DPCM), followed by statistically matched variable length coding. A buffer has been used to smooth the short term peaks in data rate, but in addition a coding algorithm is used that adapts to the rate of information generation. When the transmitter buffer is full, feedback
paths within the coder are provided to prevent buffer overflow during periods of high movement. Thus the rate of data generation is controlled by using a movement detector, temporal subsampling, spatial and element subsampling, or "stop data", i.e. if the previous procedures fail to prevent buffer overflow, data is prevented from passing to the buffer.

5.2 Types of Buffer Storage

Buffer storage can consist of magnetic tape, computer disc, or an electronic store. Whether they are built from magnetic cores or semiconductor chips, memories are subdivided into groups of bits called words. A word consists of the bits involved in each data transfer. A memory is composed of many words and each word is stored at a selected address. Data is written into the memory one word at a time and read at a later time when the information is needed. Typical word sizes for minicomputers for example, are 12 and 16 bits, and most registers are the same size as the memory word. A typical memory size is 4K words where K is equal to $2^{10}$ or 1024.

Each word in memory is given an address. Thus each memory word has two parameters, its address which locates it within the memory, and the data, which is retained within the word. Memories also require two registers, one associated with address and one with data. The memory
address register (MAR) holds the address of the word currently being accessed, and the memory data register (MDR) holds the data being written into or read out of the addressed memory location. The words stored within the memory are not immediately accessible to the system, but the (MAR) and (MDR) are available. In some memories the MAR and MDR are not part of the internal memory. In this case, they must be part of the user's circuitry. The MDR serves as a temporary storage area (buffer) for transferring words between the memory and the external system. A block diagram of basic 4 K word by 16-bit core memory is shown in Figure 5.3. The 12 bits in the MAR select 1 of the 4096 words in memory. The data transfer is between the selected word and the MDR.

The discussion presented in the buffering review established a basic understanding of the variety of buffering techniques that have been investigated, analysed, and simulated with the aim of achieving a solution to the buffering problems within teleprocessing, telemetry, and data compression systems. In this work the investigation is carried out in the data compression area, where different sources have been processed using an adaptive transform image coding-decoding system. The variable bit rate from the encoder output is passed to the variable input-fixed output rate buffer in the transmitter, through the channel and then to the fixed input-variable output rate buffer in the receiver. Different techniques are then examined to
Memory Address Register (12 bits)

Memory address (12 bits)

Memory data (16 bits)

<table>
<thead>
<tr>
<th>0</th>
<th>0 0 0 0 0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0 0 1</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 0 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>4093</td>
<td>1 1 1 1 1 1 1 1 1 1 1 1</td>
</tr>
<tr>
<td>4094</td>
<td>1 1 1 1 1 1 1 1 1 1 1 1 0</td>
</tr>
<tr>
<td>4095</td>
<td>1 1 1 1 1 1 1 0</td>
</tr>
</tbody>
</table>

Memory Data Register (16 bits)

FIG. 5.3 BLOCK DIAGRAM OF A 4K X 16 BIT MEMORY
prevent buffer underflow and overflow. These employ buffers with delay, buffers with pattern-insertion, buffers with insertion and classification map control according to buffer status, and buffers with the combination of delay with insertion and classification map control according to buffer status. In the following section the number of bits inserted, the delay value, and the buffer size are determined. The buffer size achieved with this new strategy (classification map control according to buffer status) is smaller than that suggested by Goyal et al. [116].

5.3 BUFFERING IN ADAPTIVE IMAGE PROCESSING

Assume that the instantaneous output of the adaptive variable bit rate encoder is \( R(t) \) digits per unit time. The digit stream shown in Figure 5.4 will be fed into a buffer (Figure 5.5). If the transmission rate over the channel from the output of the buffer is \( R \) (see Figure 5.6) then \( R \) must be a constant for a fixed rate channel. When the encoding algorithm is adaptive, as illustrated in the previous chapter (section 4.3), the buffering operation can suffer from two problems. These are underflow and overflow for a given buffer size.

5.3.1 The Buffer Underflow Problem [124]

The term buffer underflow means that the buffer is empty when a request is made to the buffer for information to be sent. Buffer underflow can be avoided if:

\[
\int_{0}^{t} R(t) \, dt > Rt, \quad \text{for all } t > 0 \tag{5.11}
\]
FIG. 5.4 VARIABLE BIT RATE TRANSMITTER BUFFER INPUT

FIG. 5.5 TRANSMITTER BUFFER

FIG. 5.6 FIXED BIT RATE TRANSMITTER BUFFER OUTPUT
In general the maximum transmission data rate $R$ over the channel should be equal to the long-term average of the adaptive encoder output rate $R(t)$.

$$R = \lim_{T \to \infty} \frac{1}{T} \int_{0}^{T} R(t) \, dt \quad (5.12)$$

At the same time, since $R(t)$ is variable, it can be sometimes less than the transmission bit rate, i.e. $\int_{0}^{t} R(t) \, dt$ will be less than $R_t$. During those periods when $R_t > \int_{0}^{t} R(t) \, dt$ underflow of the buffer will occur if the buffer becomes empty. In this work three possibilities are examined to avoid buffer underflow:

a) By the introduction of a delay before information is taken from the transmitter buffer.

b) A number of digits can be inserted at the transmitter buffer and deleted at the receiver buffer.

c) Underflow also can be avoided by controlling the classification map.

5.3.1.1 Delay Within The Transmitter Buffer to Prevent Underflow

If the delay method is used to overcome the underflow problem, and $DT_d$ denotes this delay, then the necessary condition for avoiding underflow is that:
\[ \int_{0}^{t} R(t) \, dt \geq R(t - DT_d), \text{ for all } t > DT_d \] (5.13)

The first term of inequality (5.13) is the total number of information samples which have been stored in the transmitter buffer up to time \( t \), and if it is called \( R_{\text{cum}}(t) \), then:

\[ R_{\text{cum}}(t) > R(t - DT_d) \] (5.14)

or,

\[ R_{\text{cum}}(t) > [Rt - R \cdot DT_d] \] (5.15)

and from (5.15) it is clear that

\[ DT_d > \frac{[Rt - R_{\text{cum}}(t)]_{\text{max}}}{R} \] (5.16)

where the term \([Rt - R_{\text{cum}}(t)]_{\text{max}}\) is the maximum difference between the total number of bits stored in the transmitter buffer and the total number of bits transmitted from it, i.e. the maximum difference between \( Rt \) and \( R_{\text{cum}}(t) \). From inequality (5.16) it is clear that the delay in the transmitter buffer must be large enough to prevent underflow for all processing operations, which will certainly increase the buffer size. The process can be shown graphically.

Figure (5.7) illustrates the delay behaviour in transmitter buffer to avoid underflow. From the figure it is clear that the delay, \( DT_d \), must be bigger than \([Rt - R_{\text{cum}}(t)]_{\text{max}}/R\) to prevent underflow. The frequent sharpness in the fluctuation of \( R_{\text{cum}}(t) \), (which depends on the source statistics), plays an important role in delay determination since the amount of delay in the transmitter buffer can differ from one source to another.
FIG. 5.7 ILLUSTRATION OF USE OF DELAY AT TRANSMITTER BUFFER TO AVOID UNDERFLOW
5.3.1.2 Pattern-Insertion to Prevent Underflow Within The Transmitter Buffer

An alternative to avoiding transmitter buffer underflow by the provision of delay is to adopt a "pattern-insertion" procedure at those instants when underflow occurs. This digit insertion mode of operation can be performed without any serious effect on the system transmission rate [124]. It is important that "pattern-insertion" systems be considered in some detail since, as will now be shown, their performance is not the same as that of delay systems. In addition, any system operating on the delay principle, but in which the delay is less than the minimum necessary to avoid buffer underflow, will have to invoke pattern-insertion at some time during the course of its operation to ensure correct operation.

The pattern-insertion mode is shown in Figure 5.8, from which it is clear that pattern-insertion removes the need for delay in the transmitter buffer to prevent buffer underflow.

In the pattern-insertion system, it is important to note exactly under what conditions insertion must occur. Insertion must begin when the transmitter buffer is empty, i.e. under the condition that the transmission rate over the channel is greater than the rate at which data enters the buffer from the binary encoder output. Insertion will then continue until the rate at which
Fig. 5.8 An illustration of pattern-insertion at transmitter buffer

(a) Input to transmitter buffer

(b) Output of transmitter buffer

- $d_1$, $d_2$, $d_3$ are digits from binary encoder output

- $n$ digits to be inserted
information enters the buffer is equal to the rate at which it is removed. If the number of digits inserted up to time \( t \) is \( d_I(t) \), the total number of digits generated by the binary encoder up to time \( t \) is \( R_{\text{cum}}(t) \), and the transmission rate is \( R \), then it follows that the insertion begins when both

\[
R_{\text{cum}}(t) + d_I(t) = R t
\]  

(5.17)

and

\[
\frac{d[R_{\text{cum}}(t)]}{dt} < R
\]  

(5.18)

But the insertion must be stopped when

\[
\frac{d[R_{\text{cum}}(t)]}{dt} = R
\]  

(5.19)

It should be noted that during periods of insertion

\[
\frac{d}{dt} [ R_{\text{cum}}(t) + d_I(t) ] = R
\]  

(5.20)

5.3.1.3 Delay In The Receiver Buffer To Prevent Distortion Within The Delay System

In adaptive coding/decoding systems, it is also necessary that buffering be provided at the receiver to convert the received constant channel rate data into the required variable rate. Just as at the transmitter buffer, a delay has to be introduced before information can be taken from the receiver buffer. This delay is necessary
to prevent a situation arising in which a request is made to the buffer for information before it actually arrives at the buffer. The effect of having too small a delay is illustrated schematically in Figure 5.9. The necessary minimum delay $DR_d$ at the receiver buffer can be determined as follows:

It is clear that the decoder buffer is receiving a constant bit stream from the channel Figure 5.9(b). When the $K$ th data sample is required for decoding, it is necessary that it be present in the receiver buffer. The $K$ th data sample arrives in the buffer $(K-1) \tau$ seconds after the arrival of the first data sample, where $\tau$ is the transmission period ($\tau = 1/R$). If the first sample is taken from the receiver buffer $DR_d$ seconds after it arrives, then the $K$ th sample is required from the buffer at time

$$DR_d + \sum_{i-1}^{K-1} T_i$$

(5.21)

where $T_i$ is the time interval between the instants at which the source generates the $i$ th and $(i + 1)$ th data digits.

If distortion of type illustrated in Figure 5.9(c) is to be avoided, then it is necessary and sufficient for $DR_d$ to have a value such that

$$DR_d + \sum_{i-1}^{K-1} T_i \geq (K - 1) \tau, \text{ for all } K > 1$$

(5.22)

In condition (5.22) the term $\sum_{i-1}^{K-1} T_i$ corresponds to the time $t$:
FIG. 5.9 ILLUSTRATES THE DISTORTION IN THE DECODER DUE TO INSUFFICIENT DELAY IN THE RECEIVER BUFFER
(a) TRANSMITTER BUFFER INPUT RATE (VARIABLE)
(b) RECEIVER BUFFER INPUT RATE (CONSTANT)
(c) RECEIVER BUFFER OUTPUT RATE (VARIABLE)
taken for the receiver to decode k-1 source digits, with
time being measured from the instant DRd at which decoding
commences. Also, k-1 is the cumulative sum Rcum(t) of
digits taken from the receiver buffer in a t second
interval. It is clear that Rcum(t) is the same cumulative
function (time delayed) as that discussed in relation to
the transmitter buffer. It thus follows that condition
(5.22) can be rewritten as

\[ DRd + t \geq \frac{R_{cum}(t)}{R} \]  \hspace{0.5cm} (5.23)

i.e

\[ R_{cum}(t) \leq (DRd + t)R \]  \hspace{0.5cm} (5.24)

and, on substituting \( t' = DRd + t \), this becomes

\[ R_{cum}(t' - DRd) \leq Rt' \]  \hspace{0.5cm} (5.25)

From condition (5.24), the delay in the receiver buffer
(DRd) can be seen to be equal to, or greater than,
\[ [R_{cum}(t) - Rt]_{max}/R \], where \([R_{cum}(t) - Rt]_{max}\) is the
maximum value of the difference between \( R_{cum}(t) \) and \( Rt \).

Condition (5.25) is illustrated in Figure 5.11, from
which it is clear that the delay DRd is determined by the
maximum extent to which \( R_{cum}(t') \) exceeds \( Rt' \), the
cumulative total of information digits received up to
that instant.
FIG. 5.11 DELAY AT RECEIVER BUFFER TO AVOID DISTORTION IN THE DECODER

$- R_{\text{cum}}(t' - DR_d)$ CURVE MUST BE ALWAYS BELOW THE CURVE $R_{t'}$ TO PREVENT DISTORTION
Because $t'$ is delayed relative to $t$ by $\Delta T_d$ (since no information arrives at the receiver until $\Delta T_d$ seconds have elapsed) it follows that the overall delay $D_{S_d}$ introduced by the system is

$$D_{S_d} = \Delta T_d + DR_d \quad (5.26)$$

5.3.1.4 Overall System Delay For The Pattern-Insertion System

In Section (5.3.1.2) pattern-insertion in the transmitter buffer to prevent buffer underflow was discussed. In this method, it is necessary to discard in the receiver buffer the inserted digits and also to introduce a delay to avoid the type of output distortion illustrated in Figure 5.9. To avoid distortion in the output at any time $t$ it is necessary to introduce a delay, say $D_{S_I}$, such that

$$D_{S_I} + \sum_{i=1}^{k-1} T_i \geq (k-1) \tau + d_I \tau \quad (5.27)$$

where

$$\sum_{i=1}^{k-1} T_i = t \quad (5.28)$$

and $k = R_{cum}(t) \quad (5.29)$

where $t$ is measured from the end of interval $D_{S_I}$

and $\tau = \frac{1}{R} \quad (5.30)$
\( d_I(t) \) corresponds to the number of digits inserted at the transmitter buffer in the period up to the instant \( t \), and \( t' \) is measured from the instant when data is first fed into the transmitter buffer. It thus follows that inequality (5.27) can be written as

\[
DS_I + t \geq \frac{R_{\text{cum}}(t)}{R} + \frac{d_I(t)}{R} \tag{5.31}
\]

i.e. \( R_{\text{cum}}(t) + d_I(t) \leq (t + DS_I) R \)

and, on substituting \( t' = t + DS_I \) this becomes

\[
R_{\text{cum}}(t' - DS_I) + d_I(t' - DS_I) \leq Rt' \tag{5.32}
\]

which must hold for all \( t' > DS_I \).

Comparing condition (5.32) and (5.25) in Section (5.3.1.3) we can see the extra term \( d_I(t' - DS_I) \) which appears in condition (5.32). This term is due to the fact that in certain cases the arrival at the receiver buffer of the \( k \)-th information digit may be delayed by an amount \( d_I(t) \) caused by the insertion of dummy digits. From a comparison of conditions (5.25) and (5.32) it is clear that for \( d_I(t' - DS_I) \) greater than zero, \( DS_I \) is greater than delay \((DR_d)\) in Section (5.3.1.3), used for the delay system only (i.e. without pattern-insertion in the transmitter buffer). The performance relating to condition (5.32) is illustrated in Figure 5.12.
FIG. 5.12 THE PATTERN-INSERTION SYSTEM WITH DELAY NECESSARY TO AVOID DISTORTION IN THE DECODER
The minimum delay \( (DS_I) \) in the pattern-insertion system can be determined from the \( d_I(t) \) function. This is achieved by the fact that digit insertion begins when both 
\[
R_{\text{cum}}(t) + d_I(t) = R(t) < R.
\]
Figure 5.13(a) shows the instants when digit insertion occurs. It should be noted that as a result of insertion the \( R_{\text{cum}}(t) \) curve is effectively coincident with the \( R(t) \) curve during periods of insertion and then goes above it. From Figure 5.13(b) it can be seen that the total number of inserted digits up to any instant in time is equal to the sum of inserted digits during all previously completed pattern-insertion periods plus the number of digits inserted up to the instant being considered in the current insertion period.

5.3.1.5 Comparison Of The Overall Delays Involved In The Two Systems

From the previous sections on the underflow problem a comparison of the overall system delay involved in the delay and pattern-insertion systems can be obtained by considering conditions (5.24), (5.31), and Figure 5.14. From condition (5.31), it is clear that, if distortion is to be avoided with the pattern insertion system, the value of delay \( DS_I \) must satisfy the condition

\[
DS_I \geq \frac{[R_{\text{cum}}(t) - R(t) + d_I(t)]}{R} \max \quad (5.33)
\]

That is to say, the necessary minimum value of the delay \( DS_I \) is determined by the maximum value of \( R_{\text{cum}}(t) - R(t) + d_I(t) \).
FIG. 5.13 ILLUSTRATION OF PATTERN INSERTION IN THE TRANSMITTER BUFFER WHERE A IS THE INSTANT AT WHICH THE INSERTION STARTS, AND B IS THE INSTANT WHERE INSERTION STOPS.
From condition (5.24) it is clear that \( \frac{[R_{\text{cum}}(t) - R_t]_{\text{max}}}{R} \) is equal to \( D_{R_d} \) which is the necessary minimum receiver delay in the delay-operated system, and from Figure 5.14 that the maximum value \( \frac{[d_1(t)]_{\text{max}}}{R} \) of \( \frac{d_1(t)}{R} \) is equal to \( DT_{d_1} \), the minimum transmitter delay in a delay-operated system.

In the delay-operated system, the minimum overall system delay, \( D_{S_d} \) is

\[
D_{S_d} = DT_{d_1} + DR_d = \frac{[R_{\text{cum}}(t) - R_t]_{\text{max}}}{R} + \frac{[d_1(t)]_{\text{max}}}{R}
\]

(5.34)

From conditions (5.33) and (5.34), it is clear that the minimum value of \( D_{S_1} \) is only equal to the minimum value of \( D_{S_d} \) in the special case in which the maximum values of \( [R_{\text{cum}}(t) - R_t] \) and \( d_1(t) \) occur at the same instant. It should be noted that this condition is satisfied if the maximum value of \( [R_t - R_{\text{cum}}(t)] \) occurs before the maximum value of \( [R_{\text{cum}}(t) - R_t] \).

One important aspect of system performance remains to be considered, and this relates to the time during which the output will contain distortion if the overall delay used in a system is less than the necessary minimum value. Assume that a system works on a delay-operated basis, for the transmitter and receiver buffers, and that the \( R_{\text{cum}}(t) \) curve is as shown in Figure 5.14(a), the minimum delay on the diagram. Let it be further assumed that the actual delay used at the transmitter buffer is \( DT_{d_1} S \), where \( DT_{d_1} S \)
is less than $DT_d$. Under these circumstances, pattern-insertion will occur in the manner shown in Figure 5.14(b). Pattern-insertion effectively forces the $R_{\text{cum}}(t)$ curve to be shifted to the dotted curve shown in Figure 5.14(a). At the receiver, let it be assumed that, instead of using the delay value of $DS_I$, which would be sufficient to avoid distortion in the output, a value of $DS_{I,S}$ shown in Figure 5.14(c) is used, where $DS_{I,S}$ is less than $DS_I$. If this is done then from 5.14(c) it can be shown that distortion will occur in the intervals A and B indicated on the Figure 5.14(c) (i.e. because the short-dashed curve is still above the curve $R_t$). In general terms, distortion will occur during those periods for which

$$R_{\text{cum}}(t - DS_{I,S}) + d(t - DS_{I,S}) > R_t \quad \text{for } t > DS_{I,S}$$

(5.35)

5.3.2 Buffer Size Estimation and The Overflow Problem

From the discussion in Sections 5.3.1.1 - 5.3.1.5 the estimation of buffer size is made at both transmitter and receiver. In the case of delay-operated systems, the transmitter and receiver buffers have to be large enough to hold a number of digits equal to the maximum value of the difference between $R_{\text{cum}}(t)$ and $R(t - DT_d)$, and $R(t + DR_d)$ and $R_{\text{cum}}(t)$, respectively. That is, the transmitter buffer size $B_{Tx}$ is given by

$$B_{Tx} = [R_{\text{cum}}(t) - R(t - DT_d)]_{\text{max}} \quad (5.36)$$
FIG. 5.14 ILLUSTRATION OF DISTORTION IN PATTERN-INSERTION SYSTEMS
and the receiver buffer size \( B_{Rx} \) by

\[
B_{Rx} = [R(t - DR_d) - R_{cum}(t)]_{max}
\] (5.37)

Now for a given transmitter buffer size \( B_{To} \), less than \( B_{Tx} \), overflow will occur, (i.e. if \( B_{To} < B_{Tx} \)) which results in data lost, and consequently distortion of the image signal. To overcome this problem the following procedures are followed. From equations (5.15), (5.28), (5.29), if the number of bits \( k_i = R_{cum}(t) \) is taken to be equal to the number of bits within a particular block of class \( i \) at a time instant \( t \) needed to encode that block of \((16 \times 16)\) samples, then

\[
R_{cum}(t) - R_t = \delta_i
\] (5.38)

where \( \delta_i \) is the increase or decrease in the buffer contents due to a block of class \( i \).

Now if the assumption is made that the transmission bit rate per element is \( b_r \), and the number of elements per block is \( n \), then \( N \), the number of transmitted bits per block is given by

\[
N = n \cdot b_r
\] (5.39)

and if \( N_{bi} \) is the number of bits to be transmitted for a block of class \( i \) (i.e. the number of bits within one of the bit allocation matrices of Figure 4.19 in Chapter IV Section 4.3.5.3), then
\[ \delta_i = N_{bi} - N \]  
(5.40)

or

\[ \delta_i = N_{bi} - n.br \]  
(5.41)

In general the buffer contents should be nearly constant, that is

\[ \sum_{k=1}^{r} \sum_{j=1}^{i} \delta_i = 0 \]  
(5.42)

where \( i = 1, 2, 3, 4 \), and \( r \) is the number of blocks of each class. Equation (5.42) is very important because if it is not met no practical buffer can be used.

If \( \sum \sum \delta_i \) is positive, then an infinitely large buffer is required. If a practical size buffer is used an overflow will occur, while if \( \sum \sum \delta_i \) is negative underflow will occur. In fact condition (5.42) is not enough to prevent overflow or underflow, because occasionally consecutive increasing or decreasing \( \delta_i \) can occur. However, the underflow problem has already been treated in earlier discussion in this chapter, and therefore the overflow problem has now to be considered.

To overcome the overflow problem, a different strategy is adopted. It is control of the classification map of the image according to buffer status. If in each frame blocks are divided into four classes according to their activity, where class four is the lowest activity class and
class one is the highest, then from equation (5.41)

\[ \delta_4 < \delta_3 < \delta_2 < \delta_1 \]  \hspace{1cm} (5.43)

The new strategy used is to control the buffer contents so that overflow will not occur. The control is achieved as follows: Two thresholds are set where the buffer is nearly full. If the buffer content reaches the first threshold, the next block class is increased by one (if possible), so that fewer bits will enter the buffer as can be seen from Equation (5.43). Let the first threshold be \( \text{Th}_1 \) and the buffer contents \( C_0 \), then

\[ \delta_i \rightarrow \delta_{i+1} \text{ if } C_0 \geq \text{Th}_1 \]  \hspace{1cm} (5.44)

where \( i = 1, 2, 3 \).

If the buffer contents reach the second threshold \( \text{Th}_2 \), then the next block class is increased by two, otherwise by only one.

If \( C_0 \geq \text{Th}_2 \) then

\[ \delta_i \rightarrow \delta_{i+2} \text{ where } i = 1, 2, \]  \hspace{1cm} (5.45a)

otherwise

\[ \delta_i \rightarrow \delta_{i+1} \text{ where } i = 3 \]  \hspace{1cm} (5.45b)

For example, the behaviour of the buffer is shown in Figure 5.15 to show the control effect.
The same strategy to control the buffer is introduced to decrease the pattern-insertion frequency. Thus a threshold $Th_3$ is set near the region of zero buffer content. If the buffer content falls below the threshold, the next block class is reduced by one (if possible) to force the system to fill the buffer faster. Thus

$$\text{if } C_0 \leq Th_3 \text{ then } \delta_i \rightarrow \delta_{i-1} \quad (5.46)$$

where $i = 2, 3, 4$.

The behaviour of the buffer is shown in Figure 5.16.

### 5.3.3 EXPERIMENTAL PROCEDURES

A variable input-fixed output bit rate buffer for an adaptive transform image coding decoding system has been simulated. Different images of size (256 x 256) elements with different statistics have been transformed and the sub-blocks categorized into four groups depending on the activity within each (16 x 16) elements block. For each group (class) the bit allocation matrix is computed, and classification maps for four classes are constructed (considering the first class as the most active). Each transformed block is normalized and then quantized using a nonuniform quantizer and the binary encoded output is fed to the variable bit rate input buffer in the transmitter shown in Figure 5.26. Five main schemes of buffer simulation have been examined:
FIG. 5.15 REPRESENTATION OF BUFFER BEHAVIOUR TO PREVENT BUFFER OVERFLOW

FIG. 5.16 REPRESENTATION OF BUFFER BEHAVIOUR TO PREVENT BUFFER UNDERFLOW
1. Buffering with delay.
2. Buffering with pattern-insertion.
3. Buffering with classification map control according to buffer status plus pattern-insertion.
4. Buffering with delay, insertion and classification map control according to buffer status.
5. Buffering with coarse quantisation to prevent overflow.

A flow chart in Figure 5.17(a) shows the simplified simulated system for the coder with the buffering scheme in general (see Appendix II). The output of the encoder is then passed to the buffer input.

5.3.3.1 The Adaptive Transform Coding System And Buffer Simulation

The design of the buffer is one of the most important tasks to be faced when implementing adaptive coding systems. Therefore in this section the buffer simulation is divided into five parts. These are:

1. The simplified transmitter buffer input.
2. The transmitter buffer output.
3. The receiver buffer input.
4. The transmitter buffer input with classification map control.
5. The receiver buffer output with variable bit rate.
PICTURE ELEMENTS

REFORMAT THE PICTURE ELEMENTS OF (256 X 256) INTO I = 256 BLOCKS OF (16 X 16) PICTURE ELEMENTS

READ DATA BLOCK I

TRANSFORM

FIND THE ACTIVITY INDEX FOR EACH BLOCK

RE-ORDER THE ACTIVITY INDEX

CATEGORIZE BLOCKS INTO FOUR GROUPS

STORE DCT COEFFICIENTS WITHIN FOUR CLASSES

YES

1<256

NO

CONSTRUCT THE CLASSIFICATION MAP

FIG. 5.14(a) TRANSMITTER FLOWCHART WITH SIMPLIFIED BUFFERING.
READ STORE $i$

FIND THE VARIANCE OF INDIVIDUAL COEFFICIENTS WITHIN THE CLASS

$\bar{x} = \frac{1}{i} \sum_{i=1}^{i} x_i$

CALCULATE STANDARD DEVIATION

FIND THE NORMALIZATION FACTOR ($c$)

FIND NORMALIZATION COEFFICIENTS

BIT ALLOCATION MATRIX OF CLASS $i$: $[M_{b,i}(u,v)]$

FIG. 5.11(c) (CONTINUED)
J = 1

READ NORMALIZED DCT OF BLOCK J, CLASS, AND MAP

FIND BIT ALLOCATION ACCORDING TO CLASS \( b \).

\[ b' \]

\[ J = J + 1 \]

CODE DCT COEFFICIENTS ACCORDING TO BIT ALLOCATION MATRIX

FORM AN 8-BIT WORD

\[ N = N + 1 \]

\[ L = L + 1 \]

STORE 8-BIT WORD IN ADDRESS \( N \) BUFFER

\[ L \neq 8 \]

\[ J = 256 \]

NEXT FRAME

NO

YES

FIG. 5.11(a) (CONTINUED)
5.3.3.1.1 The Transmitter Buffer Input

The flowchart in Figure 5.18 shows the basic operation of the simulation. The bit allocation matrix for each (16 x 16) block of coefficients from the binary encoder output is fed to the buffer. The first digit of the binary word IBI which corresponds to the number of bits IB for the (i,j)th coefficient within the (16 x 16) block has been fed to the transmitter variable rate input buffer, while the input address for the buffer is increased by one.

For the DC coefficient IBI is equal to 8 bits, and is equal to the maximum code word length in the data bit-stream. When the complete code word IBI is stored in the buffer, the following IBI which corresponds to the number of bits IB for the (i,j)th a.c coefficient is stored in the buffer. For each stored bit the input address is increased by one, and so on until a block of data is stored completely with the variable bit stream shown in Figure 5.17(b). The block is then called from the store. During this operation if the buffer input address reaches the input address maximum, then it is set to zero. It is important to note that the input address maximum must be equal to the buffer size.

5.3.3.1.2 The Transmitter Buffer Output With and Without Delay

The second part of the buffer simulation is that of the transmitter buffer output. The transmitter buffer has been
FIG. 5.17(b) ILLUSTRATES BIT STREAM FED TO BUFFER INPUT. THIS REPRESENTS THE CLASS FOUR BIT ALLOCATION MATRIX

B.S. - BIT STREAM

N.B. - NUMBER OF BITS TRANSMITTED FOR EACH ELEMENT
START
IBI - CODE WORD LENGTH
IB - NUMBER OF BITS IN IBI

K = 0

YES

K = IB?

NO

K = K + 1

BUFFER INPUT ADDRESS
INPADD = INPADD + 1

WRITE THE K-TH BIT OF THE
BINARY WORD IBI IN BUFFER
LOCATION WHOSE ADDRESS IS
INPADD

INPADD = INPADD MAX

?  

NO

YES

INPADD = 0

NEXT IBI

FIG. 5.18 TRANSMITTER BUFFER INPUT SIMULATION.
designed for first-in-first-out operation. The flowchart shown in Figure 5.19 shows the simulation of the transmitter buffer. In fact this simulation originally included no delay in the buffer, but a delay scheme was introduced to prevent or to reduce the underflow effect. If there is no delay, the first digits of the code word IBI which are already stored in the buffer start to be read at a constant bit rate, while the read address counter is incremented by one for each digit fed from the buffer location IOUTADD to the channel. However, with a system including delay, the first digit to be read from the buffer output will be delayed until the desired number of blocks have been stored in the transmitter buffer. In Figure 5.19 three blocks have been delayed.

5.3.3.1.3 The Transmitter Buffer Output With Pattern-Insertion

A pattern-insertion system to prevent underflow in the transmitter buffer has also been investigated and simulated. From the flowchart in Figure 5.20 it can be seen that pattern insertion occurs whenever the buffer content is equal to zero. Increasing the write input address counter by one causes one digit to be written into the buffer location input address, this is repeated up to a maximum of 150 digits, which have been chosen experimentally. Note that the 150 digit pattern-insertion can occur more than once during single block transmission, if this block is from a less active area of the image and the buffer is empty.
\( I = 0 \)

\( I = I + 1 \)

**START**

**NUMBER OF BLOCK \( I = 3 \)?**

**YES**

\( \text{IOUTADD} = \text{IOUTADD} + 1 \)

PASS THE VALUE STORED IN BUFFER LOCATION IOUTADD TO THE CHANNEL

FIG. 5.19 TRANSMITTER BUFFER OUTPUT WITH DELAY TO PREVENT BUFFER UNDERFLOW.
FIG. 5.20 TRANSMITTER BUFFER OUTPUT WITH INSERTION-OF 150 'ONES' TO PREVENT BUFFER UNDERFLOW.
5.3.3.1.4 The Receiver Buffer Input

Data is received from the channel at a constant rate of \( br \) bit/element and is fed to the receiver buffer. In the pattern-insertion system an inserter excluder is included in the flowchart as shown in Figure 5.21. The inserter excluder is used to delete the number of digits inserted in the transmitter buffer to prevent any underflow occurring. There is a bit counter for "ones" set in parallel to the input of the buffer as shown in Figure 5.21. If the incoming bit stream consists of 150 successive ones, then this data has to be deleted from the receiver buffer and overwritten by a new set of data in the same buffer locations. This at the same time decreases the write address counter (INRX) by an amount equal to the counter excluder, thus allowing the write address counter to rewrite the value of the received binary bit in the receiver buffer location (INRX), into the same location of the first 150 deleted 'ones'.

5.3.3.1.5 The Transmitter Buffer Input With Classification Map Control According To The Buffer Status

In Section (5.3.2) the new strategy for overcoming overflow and underflow buffering problems was discussed. The buffer behaviour has been shown graphically in Figure 5.15, and 5.16 in Section (5.3.2) for these techniques. For greater clarification the full operation is depicted in the flowchart of Figure 5.22. From the previous
FIG. 5.21 ILLUSTRATES THE SIMULATION OF THE RECEIVER BUFFER INPUT.
discussion the buffer was simulated without this technique and with a given average transmission bit rate per element, buffer size, and average coding rate.

To overcome overflow and underflow problems and to reduce the buffer size a new technique has been developed. From the flowchart in Figure 5.22 the assumption is made that the classification map has been constructed, and each block has been given its identity (class). At the beginning of transmission this technique has no effect, but during the transmission operation, when the accumulated data in the buffer reaches a certain defined threshold level Th1, this technique will start to have an effect on the system. The first and second threshold levels have been chosen from previous experiments (i.e. without classification map control). However, it is very important to note that this technique has no effect on the current block even if the buffer content reaches the first threshold level, when the next incoming block class must be increased by one. The activity class order considered is 1, 2, 3, and then 4.

Thus, for example, when class 1 is increased by one it becomes class 2 which is less active, i.e. less data will be stored in the buffer for the next block. This operation will continue as long as the buffer content is bigger than Th1. If the system is working in a busy area, then the buffer might become full within a short period of time and
overflow might occur. In this case the second threshold Th2 is set to control the classification map by, for example, moving from class 1 to class 3. Fewer bits will then be stored in the buffer, and this will certainly prevent buffer overflow. However, the buffer size must be larger than Th2 to allow the system to work in a safety condition. On the other hand the same strategy is used to control buffer underflow, when a third threshold Th3 is set. The value of this threshold must be chosen so as to reduce the pattern-insertion frequency, and consequently to prevent buffer underflow. As is shown in the controller flowchart, Figure 5.22, the value of Th3 was chosen near the empty buffering region. If the buffer content falls below Th3, the next block class is decreased by one if the number of the class i is bigger than one. This controller has been simulated using an image that can be categorized into four groups. If an image is categorized into more than four classes, the controller can be easily modified. Since we have four classes, a two bit code word can be used for the number of block class. These words are 00, 01, 10, and 11, for classes 1,2,3, and 4 respectively.

The block identity codeword is stored first in the buffer followed by the block of data that is to be transmitted. Next the number of bits IB for the first coefficient within the block of a given class needs to be obtained. Actually the first IB corresponds to the DC coefficient, which has been linearly quantized and binary
FIG. 5.22 ILLUSTRATES THE SIMULATION OF TRANSMITTER BUFFER INPUT FOR ADAPTIVE TRANSFORM CODING SYSTEMS WITH CONTROLLER.
INPADD = INPADD + 1

WRITE THE VALUE OF FIRST BIT OF \(c\) CLASS IN BUFFER LOCATION WHOSE ADDRESS IS INPADD

INPADD = INPADD + 1

WRITE THE VALUE OF SECOND BIT OF \(c\) CLASS IN BUFFER LOCATION INPADD

BLOCK OF \((I,J)\)  
\(I = 0, J = 0\)

\(I = I + 1\)

\(J = J + 1\)

OBTAIN IB  
(IB : NUMBER OF BITS CORRESPONDING TO THE \((I,J)\)-TH COEFFICIENT IN THE GIVEN CLASS)
DC COEFFICIENT

FIND NORMALIZATION COEFFICIENTS
CON = DC / 255

FIND NORMALIZED COEFFICIENTS
B'(1,1) = B(1,1)/CON

ENCODE B'(1,1) INTO AN 8-BIT BINARY WORD IB1

AC COEFFICIENT

FIND NORMALIZATION COEFFICIENTS
CON = C*(2**(IB-1))

FIND NORMALIZED COEFFICIENTS
B'(I,J) = B(I,J)/CON

QUANTIZE B'(I,J) USING NONLINEAR QUANTIZER

FIND QUANTIZER OUTPUT LEVEL IQ

ENCODE THE DECIMAL VALUE IQ INTO AN 10-BIT BINARY WORD IB1

FIG. 5.22 (CONTINUED)
$DEK = 0$

YES $K = IB$ ?

NO $K = K + 1$

BUFFER INPUT ADDRESS
$\text{INPADD} = \text{INPADD} + 1$

WRITE THE $K$-TH BIT OF THE BINARY WORD [B] IN BUFFER LOCATION WHOSE ADDRESS IS $\text{INPADD}$

$\text{INPADD} = \text{INPADD MAX}$ ?

$\text{INPADD} = 0$

---

*Fig. 5.22 (continued)*
encoded into an 8-bit binary word, IBI, followed by the a.c. coefficients using variable length code words. (These words represent the number of quantization output levels (IQ), where IQ has been binary encoded to IBI). The binary word IBI is then fed to the variable bit rate input buffer according to the number of bits IB for the (i,j)th coefficient. If IB for the (i,j)th location is zero the next coefficient must be fetched so that transmission can continue. If it is not zero the kth bit of the binary word IBI is written into the buffer and the buffer input address (INPADD) is increased by one. When the input address buffer reaches the maximum buffer size, the input address counter is set to zero.

5.3.3.1.6 Receiver Buffer Output

In adaptive decoding systems the output of the receiver buffer is at a variable bit rate, while the input is at a fixed bit rate. To overcome data distortion in the output (which has been discussed in Section 5.3.1.3) a threshold delay THR is set in the receiver buffer. In the flowchart of Figure 5.23 the simulation of the receiver buffer output is depicted with the decoding procedures for clarification. From the Figure the input data is accumulated in the receiver buffer until it reaches the threshold level THR. Then the first digit to enter the buffer is taken from the output (i.e. first in - first out). Actually, the first and second digits are the block identity or two bit block
START

NO

INRX \> THR?

YES

IOUTRX = IOUTRX + 1

TAKE THE VALUE IN Rx BUFFER LOCATION IOUTRX TO BE THE FIRST BIT IN THE 2-BITS WORD \& CLASS

IOUTRX = IOUTRX + 1

TAKE THE VALUE IN Rx BUFFER LOCATION IOUTRX TO BE THE SECOND BIT IN THE 2-BITS WORD \& CLASS

FIND THE IDCT OF THE BLOCK STORED IN THE MATRIX A

YES

I = 16?

J = 0

YES

J = 16?

NO

NO

DECODE THE WORD \& CLASS INTO CLASS

YES

BLOCK (I,J)

I = 0, J = 0

I = I + 1

J = J + 1

SET THE 8-BIT BINARY WORD IBY TO 00000000

A

B

FIG. 5.23 FLOWCHART OF RECEIVER BUFFER OUTPUT.
OBTAIN IY

IY : NUMBER OF BITS CORRESPONDING TO THE (I,J)-TH COEFFICIENT IN THE GIVEN CLASS

YES

IY = 0

\( K = 0 \)

NO

YES

K = IY

NO

IOUTRX = IOUTRX + 1

\( K = K + 1 \)

DECODE THE BINARY WORD IBY INTO THE DECIMAL VALUE IA (IA : NUMBER OF RECONSTRUCTION LEVEL ACCORDING TO IY)

YES

\( 1 + J = 2 \)

NO

DC COEFFICIENT

\( CON = DC \frac{1}{255} \)

AC COEFFICIENT

\( CON = C \cdot 2^{(IY-1)} \)

IF THE IY-TH BIT OF THE BINARY WORD IBY IS 0 SET SIGN TO +1.0, ELSE TO -1.0

\( A(1,1) = IA \cdot CON \)

\( A(I,J) = IQ(IA) \cdot CON \cdot SIGN \)

TAKE THE VALUE STORED IN Rx BUFFER LOCATION IOUTRX TO BE THE K-TH BIT IN THE BINARY WORD IBY

FIG 523 (CONTINUED).
code word. After the two code word bits have been decoded then the first incoming block class can be reconstructed. Then the data from the ith class block can begin to be taken from the buffer. To do that the 8-bit binary word IBY is set to zero and the number of bits IY, corresponding to the (I,J)th coefficient in the given class, is obtained from the bit allocation matrix for that class. The number of bits in the binary codeword IBY is taken from the buffer when counter k has been set to count the number of bits in any IBY, and at the same time the read address counter is increased by one. When the k counter is equal to the number of bits IY, then the decoding process is begun to decode the binary word IBY into the decimal value IA.

When the\((I + J)\)th location is equal to 2 at the location of the DC coefficient, the DC coefficient is linearly decoded by multiplying the decimal value IA by the normalization coefficient (CON), which gives the value of \(A(1,1)\) for the DC coefficient. Next the a.c. coefficients are taken from the buffer, decoded and reconstructed. When all the decoding operations for a block of class i are completed then it is inverse transformed and stored in matrix \(A(I,J)\), and the next block fetched and processed.

5.3.3.1.7 The Buffer Structure

a) Transmitter Buffer

The simulated transmitter buffer is shown in Figure 5.24, and consists of the transmitter buffer (Tx) which represents the buffer store and write address counter. The
FIG. 5.24 SIMPLIFIED BLOCK DIAGRAM OF TRANSMITTER BUFFER.
write address counter is clocked by clock one incrementing the counter by one for each bit as it is stored in the buffer store. The read address counter is clocked by clock 2 and is incremented by one for each bit transmitted over the channel. Both of the write and read address counters are connected to the classification map controller. In the classification map controller, the difference between the input address and output address is obtained for any time \( t \). If the difference is zero, the \( n \) bits inserter inserts 150 "ones" into the buffer during block storage. The function of the classification map control is explained in Figure 5.22 and Section 5.3.3.1.5. It is important to note that clock 1 must be synchronised with the variable data input, and clock 2 with the fixed bit rate coming from the buffer (i.e. the transmission rate).

b) Receiver Buffer

The fixed bit rate channel data is stored in the receiver buffer, as in Figure 5.25. The input buffer store (Rx) is connected to the inserter excluder in parallel. This is to exclude or to delete the number of bits inserted in the transmitter buffer to prevent transmitter buffer underflow during block transmission and is a 150 "1's" counter, i.e. if the "ones" sequence is applied to the buffer and to the counter, the counter should count 150 "1" and shift back the \( n \) bit address counter by 150.

The \( n \) bits address counter (the write address counter) is incremented by one for each incoming bit entering the
FIGURE 5.25 SIMPLIFIED BLOCK DIAGRAM OF RECEIVER BUFFER.
buffer store Rx, and the read address counter is incremented by one for each bit going out from the buffer. The read address counter is clocked by clock 2 which is synchronized with the variable bit stream of data out of the buffer, whilst the write address counter is clocked by clock 1 which must be synchronised with the data input stream.

The class bit detector, which might consist of a programmable counter and a class decoder is clocked by clock 2. The aim of the class bits detector is to determine the class number of an incoming block by using the two extra bits to determine the number of bits within the bit allocation matrix for the incoming block of the picture.

The transmitter and receiver buffers shown in Figures 5.24 and 5.25 have been used in the adaptive transform image coding and decoding system based on the previous simulation. The experimental results of this work are discussed in the following section.

5.3.4 EXPERIMENTAL RESULTS

The transmitter and receiver buffers which have been described earlier are part of the adaptive transform image coding decoding system shown in Figure 5.26. This system was first investigated and simulated without buffering procedures, and then the transmitter and receiver buffers were included. Several buffering techniques have been examined and the main approaches are as follows:
FIG. 5.26 COSINE TRANSFORM ADAPTIVE CODING-DECODING SYSTEM WITH VARIABLE INPUT-FIXED OUTPUT BIT RATE BUFFER AND CONTROLLER.
I Buffering with delay.

II Buffering with insertion.

IIIa,b Buffering with insertion and classification map control.

IVa,b Buffering with delay, insertion, and classification map control.

V Buffering with coarse quantization.

5.3.4.1 Buffering With Delay

This scheme investigates the effect of the delay in the transmitter buffer to prevent underflow, where only a few blocks of data have been stored in the transmitter buffer before the transmission is started.

Firstly two blocks of the data bit stream were stored, and then transmitted at an average transmission bit rate of one bit per element. In this case underflow occurs.

Secondly, when the number of blocks stored is increased to three, underflow does not occur for the "Girl" picture, but when the "Test Card" picture is processed and transmitted, underflow does occur. This means that the number of blocks to be delayed in the buffer can differ according to image statistics.

From Equations (5.15), (5.16) in Section (5.3.3.1.1) the number of bits that ought to be delayed in the buffer
is given by:

\[ R \cdot D T_d \geq [R_t - R_{\text{cum}}(t)]_{\text{max}} \quad (5.47) \]

where \( R \cdot D T_d \) is the number of bits that are delayed in the transmitter buffer, where this must be large enough to prevent underflow. For the "Girl" picture the number of bits delayed for the first three blocks is 1082 before the transmission is started. The number of blocks versus buffer status is shown in Figure 5.27. The Figure shows the buffer behaviour for the "Girl" picture with a delay of three blocks (the delay appears on the Figure at the start of the curve on the X axis). This delay was enough to prevent underflow for this processed picture, at an average rate equal to 0.98 bits per element and including the two code word bits for each block class as overhead information. The maximum buffer size is shown on the figure and is approximately equal to 2424 bits.

When the "Test card" picture is processed with a three block delay, underflow occurs. The number of bits for these three blocks is 513, and therefore insertion is included into the system to prevent underflow. The buffer behaviour for the processed "Test card" picture with a generated average bit rate equal to 0.95 bits/element and an average transmission bit rate equal to one bit per element is shown in Figure 5.28. Pictures of the original and processed images are shown in Figures 5.29 and 5.30 respectively. From Figure 5.28 it can be seen that the maximum buffer size for the "Test card" picture is equal
FIG. 5.27 ILLUSTRATES THE BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH DELAY ONLY FOR THE "GIRL" PICTURE
FIG. 5.2B ILLUSTRATES THE BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH DELAY ONLY FOR THE "TESTCARD" PICTURE
FIG. 5.29 THE ORIGINAL PICTURES
FIG. 5.30 RESULTS OF SCHEME I
to 2120 bits.

In the receiver buffer, distortion can be avoided with four blocks delay for the "Girl" picture, and six blocks delay for the "Test card" picture.

5.3.4.2 Buffering With Pattern-insertion (Scheme II)

In this scheme, to prevent buffer underflow, a number of "ones" is inserted whenever the content of the buffer is equal to zero (i.e. when the buffer is empty). During the time interval required to transmit the inserted bits, binary digits produced by the source are buffered and read out after the entire number of bits inserted has been transmitted. The insertion frequency depends upon the relationship between the transmission bit rate and the average compression bit rate. If the transmission bit rate is much higher than the average compression rate, then the number of pattern-insertion sequences becomes high. The pattern-insertion should be stopped when the value of \( \frac{d[R_{\text{cum}}(t)]}{dt} = R \) (i.e. the slope of \( R_{\text{cum}}(t) \)) is equal to the transmission bit rate.

In the receiver buffer, to avoid distortion, four blocks are delayed for the "Girl" picture (1299 bits) and six blocks for the "Test card" picture (i.e. 1626 bits). The transmitter buffer behaviour is depicted in Figures 5.31 and 5.32 for the "Girl" and "Test card" pictures respectively. The processed pictures are shown in Figures
FIG. 5.31 ILLUSTRATES THE BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH PATTERN-INSERTION ONLY FOR THE "GIRL" PICTURE.
FIG. 5.32 ILLUSTRATING THE BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH PATTERN-INSERTION ONLY FOR THE TESTCARD PICTURE
FIG. 5.33 RESULTS OF SCHEME II
5.33(a) and 5.33(b). The total number of bits which have been inserted to prevent buffer underflow is equal to the number of bits delayed in scheme I for both the "Girl" and "Test card" pictures.

5.3.4.3 Buffering With Insertion And Classification Map Control According To Buffer Status

Scheme IIIa

The theoretical and simulated principles of this scheme have been discussed in Sections (5.3.2) and (5.3.3.1.5) where the classification map of the image is controlled according to the buffer status. The reason for this control is to prevent both overflow and underflow of the buffer. For that purpose, the three thresholds Th₁, Th₂, Th₃ are set equal to 2500, 3000, and 256 respectively. In this case Th₃ is present to prevent buffer underflow, but the other two thresholds Th₁ and Th₂ initially have no effect on the system for preventing overflow. The overflow control operation in this scheme can be seen when the content of the buffer reaches either Th₁ or Th₂. Therefore, the dynamic range of the controller (Th₃ ↔ Th₁ ↔ Th₂) has been changed, and operating values of Th₁ and Th₂ have been chosen with consideration of the buffer size in Scheme II, and set equal to (1500, 2000) bits, while Th₃ is still the same. In this case, when the buffer content reaches the threshold Th₁ (1500 bits) the block class i is increased by one
(i.e. from class 1 to class 2 for example, where class 1 is the more active) to prevent the overflow and as before for threshold $\text{Th}_3$ (256 bits) to prevent the underflow. Now let us assume that the coder is working in a long term active area. Thus, the buffer input will be fed by an active bit stream, which might result in buffer overflow if the content of the buffer is higher than $\text{Th}_1$ (1500 bits). To make this system work in a more safe condition threshold $\text{Th}_2$ is set equal to 2000 bits to increase the block class by two if it is possible (i.e. class 1 changes to class 3, for example). Two bit code words are sent for each block class at the beginning of the data block transmission. In the receiver, the classification map is reconstructed from the block class code word as before. The maximum buffer size as seen from the buffer behaviour Figures 5.34 and 5.35 for the "Girl" and "Test card" pictures respectively is equal to (1640, 1710) bits. In this scheme the threshold $\text{Th}_2$ does not affect the system. However, compared with that of the previous scheme the buffer size is reduced from 2022 to 1640 bits for the "Girl" picture. The results are shown in Figure 5.36(a,b).

**Scheme IIIb**

A further investigation into this method has been made in an attempt to assign threshold $\text{Th}_1$ as low as possible. The purpose of this is to reduce the buffer size, forcing the controller to work under difficult conditions, and to
FIG. 5.34 ILLUSTRATING THE BUFFER STATUS VERSUS NUMBER OF BLOCKS
WITH INSERTION AND CLASSIFICATION MAP CONTROL ACCORDING
TO BUFFER STATUS FOR THE "GIRL" PICTURE
FIG. 5.35 ILLUSTRATING THE BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH INSERTION AND CLASSIFICATION MAP CONTROL (ACCORDING TO BUFFER STATUS) FOR THE TESTCARD PICTURE.
FIG. 5.36 RESULTS OF SCHEME IIIa
access the flexibility of the system. Firstly, \( T_h \) is set to 1000 bits and \( T_2 \) to 1500 bits. In this case when the buffer content reaches the \( T_h \) threshold, the controller will change the class \( i \) to class \( i + 1 \). Referring to Scheme IIIa where the maximum buffer size is equal to 1640 bits and 1710 bits for the "Girl" and "Test card" images respectively, the classification map class \( i \) will change in the range between (1000-1640) bits for the "Girl" image and between (1000 - 1710) for the "Test card" image. In fact, the efficiency of the controller only allows the buffer size to reach a maximum of 1274 bits for the "Girl" picture, and it does not reach the value of \( T_2 \) (1500) bits, as shown in Figure 5.37(a). Figure 5.37(b) shows the buffer behaviour for the "Test card" picture where the maximum buffer size is equal to 1261 bits. Note that this scheme utilises the insertion process. However, insertion in Figure 5.37(a) occurs once only, and in Figure 5.37(b) does not occur at all. The results for the two transmitted pictures are shown in Figure 5.38(a,b).

5.3.4.4 Buffering With Delay, Insertion And Classification

MAP Control According To Buffer Status

Scheme IVa

The combination of delay, pattern-insertion, and classification map control has been used in this scheme to prevent overflow and underflow of the buffer. Three blocks of data are delayed in the transmitter buffer before transmission is started. Insertion is included in this
FIG. 5.37(a) ILLUSTRATING THE BUFFER STATUS VERSUS NUMBER OF TRANSMITTED BLOCKS WITH INSERTION AND CLASSIFICATION MAP CONTROL ACCORDING TO BUFFER STATUS FOR THE "GIRL" PICTURE
FIG. 5.37(b) ILLUSTRATING THE BUFFER STATUS VERSUS NUMBER OF TRANSMITTED BLOCKS WITH INSERTION AND CLASSIFICATION MAP CONTROL ACCORDING TO BUFFER STATUS FOR THE TESTCARD PICTURE.
FIG. 5.38 RESULTS FOR SCHEME IIIb
scheme to prevent buffer underflow during data block transmission if the content of the buffer is equal to zero. The dynamic range of the controller to control the classification map is set to (256, 1500, 2000) bits for thresholds $Th_3$, $Th_1$, $Th_2$ respectively, where the maximum buffer size reaches 1825 bits for the "Girl" picture and 1716 bits for the "Test card" picture. In the receiver the number of blocks delayed is the same as in scheme II to prevent image distortion. Buffer behaviour is shown in Figures 5.39 and 5.40 and the transmitted images using the buffer are shown in Figure 5.41(a) and 5.41(b) for the "Girl" pictures and "Test card" respectively.

**Scheme IVb**

In this scheme, unlike the previous schemes, the influence of $Th_2$ has been examined. Therefore, the range between $Th_1$ and $Th_2$ is decreased and set equal to (1400-1500) bits. Therefore the effect of $Th_2$ can be seen where the number of classes is changed from $i$ to $i+2$, if possible (for example, block class 1 has been changed to class 3) when the buffer content reaches the value of $Th_2$ (1500) bits. This scheme is very important and it does operate very well. However, the buffer size is increased in this scheme compared with Scheme IIIb. The buffer behaviour is illustrated in Figure 5.42 and 5.43 for the "Girl" and "Test card" pictures respectively, while the transmitted pictures are displayed in Figure 5.44(a,b).
FIG. 5.39 ILLUSTRATING THE VARIABLE INPUT FIXED OUTPUT BIT RATE BUFFER
STATUS VERSUS NUMBER OF PROCESSED BLOCKS WITH DELAY, INSERTION
AND CLASSIFICATION MAP CONTROL FOR THE "GIRL" PICTURE
FIG. 5.40 ILLUSTRATING VARIABLE INPUT-FIXED OUTPUT BIT RATE BUFFER
STATUS VERSUS NUMBER OF BLOCKS WITH DELAY, INSERTION, AND
CLASSIFICATION MAP CONTROL FOR THE "TESTCARD" PICTURE
FIG. 5.4 RESULTS FOR SCHEME IVa
FIG. 5.42 ILLUSTRATING VARIABLE INPUT-FIXED OUTPUT BIT RATE BUFFER STATUS VERSUS NUMBER OF TRANSMITTED BLOCKS WITH DELAY, INSERTION AND CLASSIFICATION MAP CONTROL FOR THE "TESTCARD" PICTURE
FIG. 5.43 ILLUSTRATING THE VARIABLE INPUT-FIXED OUTPUT BIT RATE BUFFER STATUS VERSUS NUMBER OF TRANSMITTED BLOCKS WITH DELAY, INSERTION AND CLASSIFICATION MAP CONTROL FOR THE "GIRL" PICTURE
FIG. 5.44(a,b) RESULTS OF SCHEME VIb FOR THE "GIRL" AND "TESTCARD" PICTURES
The maximum buffer size in this scheme is equal to (1587, 1524) bits for "Girl" and "Test card" pictures respectively. The insertion included in Schemes VIa and VIb does not in fact occur as can be seen from the graphs (5.39-40, 5.42-43).

5.3.4.5 Buffering With Coarse Quantization

Unlike the previous strategy, where classification map control has been used to prevent buffer underflow and overflow, a different strategy is used here, where coarse quantization is used to prevent buffer overflow. However, to prevent underflow in this method the same strategy used in Scheme I and II is still used here (i.e delay and insertion). The main idea in this method is to reduce the number of quantization levels when the buffer becomes full. Actually, the buffer becomes full when the picture information in a particular region is too high, or in highly detailed regions. The reduction of the number of quantization levels at this instant produces an encoder which degrades more gracefully than in the previous schemes when the picture information rate is high. Figure 5.45 is part of a block diagram of the Adaptive transform image coding system, (Figure 5.26 Section 5.3.4.1) in which buffer overflow is avoided. This is achieved by controlling the bit allocation matrices of the four image classes [see Figure 4.19(a,b)]. To avoid buffer overflow a threshold (Th) is set near to the buffer "full" state.
When the buffer status is less than the defined threshold, the encoder operates normally and there is no control of the allocation matrix. When the buffer content is equal to or greater than $Th$, the buffer status is sent to the controller to switch $S1$ from position $Q_n$ to $Q_{n-1}^*$ where $n$ must be greater than one. At the same time the controller sends a flag to control the bit allocation matrix for the incoming block. The bit allocation matrix is treated as follows.

If the number of bits $n$ is assigned to the $(i,j)$th coefficient within the incoming block and the buffer is full, then the controller forces the bit allocation matrix to become:

$$n(i,j) = n(i,j) - 1, \text{ for } n > 1$$  \hspace{1cm} (5.47)

From Equation (5.47) it is clear that the number of bits per incoming block is reduced, which gives a decrease in the number of quantizer levels (i.e. $2^7$ levels changes to $2^6$ for example), and consequently, since one bit is removed from the buffer at each sample time the buffer content in this condition will either decrease or remain the same. This gradually decreases the number of bits in the buffer.

When the buffer content decreases to the predetermined threshold value, the buffer is declared normal and a buffer status signal causes the system to switch back to normal operation (i.e. position $Q_n$ of switch $S_1$).
FIG. 5.45 ILLUSTRATES QUANTIZATION LEVEL REDUCTION BY CONTROLLING THE BIT ALLOCATION MATRIX OF THE INCOMING BLOCK WHEN THE BUFFER IS FULL.

* - FOR n > 1
FIG. 5.46 BUFFER BEHAVIOUR WITH THE COARSE QUANTIZATION METHOD FOR THE "GIRL" PICTURE. BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH 3 BLOCK DELAY BEFORE TRANSMISSION IS STARTED. THRESHOLD USED IS EQUAL TO 2000.
BUFFER STATUS

FIG. 5.47 BUFFER BEHAVIOUR WITH THE COARSE QUANTIZATION FOR THE "TEST CARD" PICTURE. BUFFER STATUS VERSUS NUMBER OF BLOCKS WITH 3 BLOCK DELAY BEFORE TRANSMISSION IS STARTED. THRESHOLD USED IS EQUAL TO 1700 BITS.
FIG. 5.48 RESULT OF USING COARSE QUANTIZATION TO PREVENT BUFFER OVERFLOW
(a) ORIGINAL PICTURE
(b) PROCESSED PICTURE
FIG. 5.49 RESULT OF USING COARSE QUANTIZATION TO PREVENT BUFFER OVERFLOW
(a) ORIGINAL PICTURE
(b) PROCESSED PICTURE
When Equation (5.47) is applied to the encoder, the decoder should be informed, and therefore a flag signal must be sent to the receiver. The decoder will copy the encoder for this operation.

The experimental results for this method are shown in Figures 5.46-49. Figure 5.46 shows buffer behaviour for the "Girl" picture with threshold value Th equal to 2000, while for the "Test card" it is set equal to 1700 as in Figure 5.47. From Figure 5.46 it can be seen that the buffer size reaches 2071 bits, and from Figure 5.47 the buffer size reaches 1721 bits. From this, the decision is made, that the value of Th cannot be equal to the maximum buffer size and therefore the maximum buffer size should be increased above the value of Th to prevent buffer overflow. A comparison between these two figures (5.46 and 5.47) and the figures for Scheme I (5.27 and 5.28) shows that a reduction of buffer size will occur using bit allocation matrix control. If this is not used the buffer size will be equal to the buffer size in Scheme I. The transmitted pictures are shown in Figure 5.48 and 5.49. Both the transmission bit rate and the average coding rate which are used in this method are the same as those used for the classification map control strategy.
5.3.5 DISCUSSION

A variable bit rate input-fixed bit rate output buffer for adaptive transform image coding-decoding systems has been investigated in this chapter. The experimental results which have been presented are for an adaptive transform image coding-decoding system using four image categories (four classes) of activity within an image of 256 x 256 elements which is divided into 16 x 16 blocks, each block consisting of 16 x 16 elements. These four classes define the image classification map.

Five buffering schemes have been investigated. In Schemes I and II, the underflow problem has been studied and it is concluded that to prevent underflow pattern-insertion (Scheme II) and data block delay (Scheme I) can be used. However, Schemes I and II can be combined to prevent buffer underflow. The more complicated question in the buffering area is the overflow problem and in Schemes IIIa,b and IVa,b a new strategy is applied to try to solve this problem. The new strategy used is to control the classification map of the image according to buffer status, without any interference with the coding algorithm (as in [111]), to prevent buffer overflow when the buffer becomes full. In fact this strategy may produce noise in the high activity regions of the picture (where it is least-disturbing to the viewer) when the controller increases class i to class i+1 (where class i+1 is less active). On the
other hand picture quality can be improved when the buffer becomes empty by decreasing class \( i \) to class \( i-1 \) (where class \( i-1 \) is more active). However, with high frequency of increasing and decreasing class \( i \), the efficiency of the coding algorithm can be degraded. Therefore some compromise solution should be employed. Thus, in Scheme VIb, \( \text{Th}_1 \) and \( \text{Th}_2 \) are chosen close together (1400, 1500 bits) taking into consideration the maximum buffer size of Scheme I. Table 5.IIIa,b illustrates the performance of the system for Scheme IVb. From Table 5.IIIa the controller behaviour for "Test card" picture can be seen. For example, class 4 goes to class 3 fifteen times to prevent buffer underflow, and overall the classes change in total 40 times. In Table 5.IIIb to prevent overflow with a given controller dynamic range, class 1 goes to class 2 four times, and to class 3 once only. Of the 256 blocks within the classification MAP 15% have been changed up and down in terms of activity to prevent underflow and overflow of the buffer. The increase of the average compression rate to prevent buffer underflow is 0.05 bits/element which in turn gives an improvement of the picture quality since this fraction of bits has been added by the controller as image information instead of overhead transmission over the channel. The dynamic range of the controller (256, 1400, 1500) and buffer size are shown in Table 5.I,(Scheme IVb for this example).

The system performance with classification map control can be indicated by another parameter, this being the
buffer size. If the buffer size is chosen to be as small as possible compared with the buffer size in Scheme I, the system performance can be observed when the buffer is full or empty. Therefore the buffer sizes were chosen relatively small for Scheme IIIa,b and IVa,b. From Table 5.I the results can be seen for four schemes with different buffering strategies. From Scheme I where the buffer size is maximum (2424 bits) and Scheme IIIb where the buffer size is minimum (1274 bits) for the "Girl" picture, one can evaluate the performance of the system with regard to the picture quality and average mean square error (Table 5.II). The average mean square error in Table 5.II for the "Girl" picture for Scheme I is equal to 0.10%, while for the same picture with Scheme IIIb it is equal to 0.14%, which is still acceptable (see Figures 5.30 and 5.38 for picture subjective test). So the reduction in buffer size compared with Scheme I is about \( \frac{1}{2} \), with relatively small increase in average mean square error of the image.

In Schemes IIIa,b and IVa,b different controller dynamic ranges for the "Girl" and "Test card" pictures have been set for the purpose of evaluating this system in more difficult conditions to prevent underflow and overflow. In Scheme IIIa and IIIb, where the insertion and classification map control has been studied, the effectiveness of the different dynamic ranges produces a clear reduction in buffer size, but with a relatively high mean square error for the "Test card" picture (Table 5.II). This results from the high
frequency fluctuation of the controller (because $T_h_1$ is relatively low). Also blurred areas in high activity blocks do appear with Scheme IIIb Figure 5.38. In Schemes IVa and IVb the effectiveness of the controller dynamic range is more important, because the buffer size reaches threshold $T_h_2$ where class $i$ goes to class $i+2$ (class $i+2$ is less active than class $i$ by two steps). In this scheme delay, plus insertion and classification MAP control are used. The buffer size is reduced compared with that in Scheme I without any noticeable image degradation [Figure 5.44(a,b)]. However, in Scheme IVa,b the number of blocks delayed for the overall system is higher than that for Scheme IIIa,b as can be seen from Table 5.IV. However, in Scheme IVb, the overall system delay is less than that in Schemes I and II.

The coarse quantization method used to prevent buffer overflow is in fact less complicated than the classification map control strategy, but the latter is a more efficient technique in terms of preventing the overflow problem. A comparison of buffer sizes between these two strategies shows that they are nearly the same. The advantage of classification map control is derived from the ability to decrease the buffer content more efficiently than is the case with coarse quantization unless another procedure is to be used with this latter technique. The graceful degradation of the image quality with coarse quantization is, however, an advantage compared with classification map control. With
Table 5.1 Buffer size for four schemes with different controller dynamic ranges and different image sources

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Controller dynamic range</th>
<th>Buffer size</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Th₃</td>
<td>Th₁</td>
</tr>
<tr>
<td>I with delay only</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>II with insertion</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>IIIa with insertion + MAP control</td>
<td>256 ←→ 1500 ←→ 2000</td>
<td>1640</td>
</tr>
<tr>
<td>IIIb with insertion + MAP control</td>
<td>256 ←→ 1000 ←→ 1500</td>
<td>1274</td>
</tr>
<tr>
<td>IVa with insertion + delay + MAP control</td>
<td>256 ←→ 1500 ←→ 2000</td>
<td>1825</td>
</tr>
<tr>
<td>IVb with insertion + DELAY + MAP control</td>
<td>256 ←→ 1400 ←→ 1500</td>
<td>1587</td>
</tr>
<tr>
<td>Method</td>
<td>Scheme</td>
<td>Normalized Mean Square Error %</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Girl Picture</td>
</tr>
<tr>
<td>Delay</td>
<td>I</td>
<td>0.10</td>
</tr>
<tr>
<td>Insertion</td>
<td>II</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>IIIa</td>
<td>0.13</td>
</tr>
<tr>
<td>Classification</td>
<td>IIIb</td>
<td>0.14</td>
</tr>
<tr>
<td>MAP</td>
<td>IVa</td>
<td>0.13</td>
</tr>
<tr>
<td>Control</td>
<td>IVb</td>
<td>0.19</td>
</tr>
<tr>
<td>Coarse Quantization</td>
<td>-</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 5.II  Average normalized mean square error for "Girl" and "Testcard" pictures for the four schemes illustrated in Table 5.1 and the coarse quantization method. The differences in NMSE between these schemes are caused by the buffer controller.
Table 5.11a,b  Scheme IVb with the "Test card" picture

<table>
<thead>
<tr>
<th>Controller behaviour to prevent buffer underflow</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of bits in a class</td>
<td>Class No.</td>
<td>New Class No.</td>
<td>No. of Changes</td>
<td>No. of bits</td>
<td>Total number</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>increased each</td>
<td>of bits added per</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>time</td>
<td>class</td>
</tr>
<tr>
<td>73</td>
<td>4</td>
<td>3</td>
<td>15</td>
<td>152</td>
<td>2280</td>
</tr>
<tr>
<td>225</td>
<td>3</td>
<td>2</td>
<td>11</td>
<td>69</td>
<td>759</td>
</tr>
<tr>
<td>295</td>
<td>2</td>
<td>1</td>
<td>14</td>
<td>73</td>
<td>1022</td>
</tr>
<tr>
<td>367</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>40</td>
<td></td>
<td>4061</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Controller behaviour to prevent buffer overflow</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of bits in a class</td>
<td>Class No.</td>
<td>New Class No.</td>
<td>No. of Changes</td>
<td>No. of bits</td>
<td>Total number</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>decreased</td>
<td>of bits subtracted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>per class</td>
</tr>
<tr>
<td>367</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>77</td>
<td>292</td>
</tr>
<tr>
<td>295</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>69</td>
<td>207</td>
</tr>
<tr>
<td>225</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>152</td>
<td>152</td>
</tr>
<tr>
<td>73</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>142</td>
<td>142</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td>793</td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td></td>
<td></td>
<td>19% of all blocks</td>
<td>0.05* bits/pel</td>
</tr>
</tbody>
</table>

* 0.05 is equal to [(4061-793)/65536] bits/pel
Table 5.IV  Delay used in Schemes I, II, III and IV.

<table>
<thead>
<tr>
<th>No. of Scheme</th>
<th>Delay (in blocks)</th>
<th>Overall System Delay (in blocks)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>At Tx buffer</td>
<td>At Rx buffer</td>
</tr>
<tr>
<td></td>
<td>GP</td>
<td>TCP</td>
</tr>
<tr>
<td>I</td>
<td>3B</td>
<td>3B + INS(\equiv 2B)</td>
</tr>
<tr>
<td>II</td>
<td>INS.</td>
<td>INS.</td>
</tr>
<tr>
<td>IIIa</td>
<td>INS.(1)</td>
<td>INS.(NONE)</td>
</tr>
<tr>
<td>IIIb</td>
<td>INS.(1)</td>
<td>INS.(NONE)</td>
</tr>
<tr>
<td>IVa</td>
<td>3B + INS.(NONE)</td>
<td>3B + INS.(NONE)</td>
</tr>
<tr>
<td>IVb</td>
<td>3B + INS.(NONE)</td>
<td>3B + INS.(NONE)</td>
</tr>
</tbody>
</table>

Definitions:

INS - Pattern-insertion
INS.(1) - Insertion occurs once only
INS.(NONE) - Insertion does not occur
3B - Three blocks delay (for example)
INS\(\equiv 2B\) - (Insertion occurs where the number of bits inserted is equal to 2 blocks of class 1.)
GP - The "Girl" picture
TCP - The "Testcard" picture

* - One extra block has been delayed in the receiver buffer as a safety condition.
these various advantages and disadvantages the designer of an adaptive transform image coding-decoding system can perhaps choose a suitable strategy in the buffering area to solve the problems of underflow and overflow.

5.3.6 Conclusion

The image classification map control strategy should be used to prevent buffer underflow and overflow when the buffer is designed for adaptive image coding-decoding systems. With this method pattern-insertion must be included in the system to prevent buffer underflow during a block transmission time interval. To reduce the insertion frequency a few blocks of data can be delayed at the beginning of transmission. From Table I the optimum buffer size is achieved in Scheme IVb which is suggested for use in practical applications, and a RAM size of 4048 bits is suitable for use as a buffer.

The coarse quantization method used is seen to be a promising method, also, to prevent buffer overflow. The buffer size that can be used with this method is nearly the same as for the classification map control strategy.
CHAPTER VI

ADAPTIVE TRANSFORM CODING
OF
MOVING IMAGES
WITH
VARIABLE BIT RATE BUFFERING
6.1 **INTRODUCTION**

In Chapter IV a fully adaptive transform coding-decoding system for still images was investigated, where the variable bit rate output of that system can be smoothed using the buffering techniques discussed in Chapter V.

In this chapter a moving image sequence has been processed using the same basic strategy discussed in Chapters IV and V with modification of the bit assignment algorithm appropriate to moving sequences to improve subjective picture quality and to achieve approximate bit rate control over the channel. Five schemes have been investigated which will be discussed later in this chapter.

Generally, video processing starts with a two-dimensional distribution of light intensity. Three dimensional scenes must first be projected onto a two-dimensional plane by an optical imaging system, where, if moving objects are to be accommodated, the light-intensity distribution is then usually raster scanned, (in television the scene is repetitively raster scanned with interlace to avoid the flicker effect) to produce 25 to 30 frames per second. Each frame is composed of many lines, e.g. 525 (Japan, North America), 625 (elsewhere), 250 for video telephone, and over 2000 lines for documents and photographs, which will be equivalent, in frequency, to zero to 4 or 6 MHz, with each scan line having spatial frequency components up to about 150 cycles or more.
per picture width. Photographs and documents require 500 to 2000 cycles per scan line. The advantages of digital representation of information over the corresponding analogue form have been discussed in Chapter II Section 2.1 and many researchers have been involved in achieving these advantages with image processing systems for moving sequences. One example is first order interframe differential pulse code modulation (DPCM) coding where the small differences between frame elements are quantized and coded for transmission [40]. Conditional replenishment has also been used, where the unchanged areas of the image are repeated at the receiver and the moving areas coded and transmitted to update the stored picture at the receiver. With this method, the data rate has been reduced to 1 bit per sample for video telephone pictures [42]. Motion in the scene has also been taken into account, where prediction of the current element value in the present frame is made from the corresponding element of the previous frame by following the element along the motion trajectory, rather than along the temporal axis. Using the motion of an element in this way in image coding is referred to as motion compensation. A number of algorithms have been published by several researchers [43,88,98], mainly restricted to two-dimensional motion and in particular to simple translation. Recently, various algorithms for motion estimation and their application to adaptive hybrid coding have been studied in order to improve the accuracy of estimation or to reduce the computation required [99]. In the present chapter, the principles
of intraframe and interframe coding will be briefly discussed, followed by descriptions of experimental work and the results obtained.

6.2 PRINCIPLES OF INTRAFRAME AND INTERFRAME CODING

Interframe coding is based upon the main idea that moving image sequences contain a great deal of frame-to-frame redundancy because picture areas are scanned every frame whether they have changed or not. The unchanged portion of the signal need not be transmitted every frame, but the signal describing the moving region must be transmitted, of course, and it requires progressively less fidelity as motion increases. Several techniques have been proposed, and can be categorised into three groups which are 1) predictive interframe coding, 2) intraframe transformation, and DPCM in the temporal direction, 3) three-dimensional transform coding. At the output of these systems the data rate may be variable and important factors which should be considered are smoothing of the data flow when it is generated at an irregular rate, and methods for correcting transmission errors.

6.2.1 Coding Of Interframe Images

Various simple techniques have been used for interframe coding, such as conditional replenishment (which may include cluster coding where the picture is segmented into changed and unchanged areas), and spatial and
temporal resolution exchange by subsampling [126]. The average rate achieved for video telephone signals is about one bit per picture element with a buffer size of 67000 bits to smooth the output of the encoder [100]. Conditional replenishment is particularly relevant here, and will be briefly examined next.

6.2.1.1 Conditional Replenishment

This technique is based on a simple method of detection and coding of moving areas which are replenished from frame to frame. If \( f(i,j,k) \) denotes the frame or field element at location \( (i,j) \) in frame \( k \), then the interframe difference signal is

\[
e(i,j,k) = f(i,j,k) - \hat{f}(i,j,k-1)
\]  

(6.1)

where \( \hat{f}(i,j,k-1) \) is the reproduced value of \( f(i,j,k-1) \) at \((k-1)th frame. If the magnitude of \( e(i,j,k) \) is greater than a predetermined threshold then it is quantized and coded for transmission. At the receiver, an element is reconstructed either by repeating the value of the element at the same location in the previous frame if it came from a stationary area or by replenishment from the decoded difference signal if it came from a moving area, i.e.

\[
\hat{f}(i,j,k) = \begin{cases} 
\hat{f}(i,j,k-1) + e(i,j,k), & \text{if } |e(i,j,k)| > TH \\
\hat{f}(i,j,k-1) & \text{otherwise}
\end{cases}
\]

(6.2)
where $\theta$ is the predetermined threshold value.

6.2.2 Intraframe Transform Coding with DPCM in the Temporal Direction

The hybrid transform and DPCM coding technique was proposed by Habibi [129] and employs a one-dimensional unitary transform coder cascaded with parallel DPCM coders. The principles of hybrid transform and DPCM coder have been extended to include interframe coding of image sequences, where the advantage of such techniques derives from the correlation between consecutive frames as well as in the spatial domain to achieve further reduction in channel bandwidth requirements [130].

In this coding technique, a two-dimensional unitary transform is performed on spatial sub-blocks within each frame. One of the bank of parallel DPCM linear prediction coders is then applied to each set of transform coefficients in the temporal direction. The resulting sequences of transform coefficient temporal differences are then quantized and coded for transmission. Image reconstruction takes place at the receiver where the transform coefficient differences are decoded, and a replica of each transmitted image is reconstructed by two-dimensional inverse transformation.

The intraframe coding technique has been discussed in Chapter IV for still images using the discrete cosine transform (DCT). In the case of hybrid coding, if the DCT is
chosen then the transformation is given by

\[ T(u,v,k) = \frac{2}{N} C(u) C(v) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x,y,k) \cos\left(\frac{\pi}{N}(x+\frac{1}{2})\right) \cos\left(\frac{\pi}{N}(y+\frac{1}{2})\right) \]  

(6.3)

and the inverse cosine transform (IDCT) is

\[ f(x,y,k) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u,v,k) \cos\left(\frac{\pi}{N}(x+\frac{1}{2})\right) \cos\left(\frac{\pi}{N}(y+\frac{1}{2})\right) \]  

(6.4)

where \( C(0) = \frac{1}{\sqrt{2}} \), \( C(1), C(2) \ldots \ldots C(N-1) = 1 \), and \( k \) is the frame number. Having two-dimensional transformation of the \( k \)th frame, at each spatial frequency \( (u,v) \), the DPCM coder quantizes and codes the difference signal between temporally adjacent transform coefficients defined by

\[ D(u,v,k) = T(u,v,k) - \hat{T}(u,v,k) \]  

(6.5)

where \( \hat{T}(u,v,k) \) is the transform coefficient value predicted from the previous frame [the \((k-1)\)th frame] given by

\[ \hat{T}(u,v,k) = a_1(u,v) T(u,v,k-1) + a_0(u,v) \]  

(6.6a)

where

\[ a_1(u,v) = \frac{E\{ T(u,v,k) \cdot T(u,v,k-1) \}}{E\{ T^2(u,v,k-1) \}} \]  

(6.6b)

is the optimum weighting prediction coefficient.
{i.e., it is chosen to minimise the mean-square prediction difference $\mathbb{E}[D^2(u,v,k)]$. With the assumption of a zero mean random process in each frame $a_0(u,v) = 0$.

6.2.3 Three-Dimensional Transform Coding

In this technique the image sequence is segmented in three-dimensions (for example a $16 \times 16 \times 16$ block) and then the three-dimensional transformation is applied (see Equation 3.8b in Chapter III). In adaptive coding systems, the block of three-dimensional transform coefficients is adaptively coded and transmitted. At the receiver the transmitted data is decoded and inverse transformed (see Equation 3.8c, Chapter III) to reconstruct the original image sequence. A serious drawback of transform interframe coders is the requirement for multiple-frame storage of the transform coefficients, and this characteristic of three dimensional transform coders severely limits their usefulness in practical interframe coding systems.

6.3 ADAPTIVE INTRAFRAME TRANSFORM CODING SYSTEM FOR MOVING IMAGE SEQUENCES

6.3.1 System Description

The adaptive transform coding system described in Chapter IV Section 4.3.5.1 (Figure 4.8) has been extended to the processing of moving image sequences. The picture
used in the simulation is a head and shoulders scene consisting of eight frames each of which has 64 x 96 picture elements and is shown in Figure 6.1. Each frame is divided into a number of 16 x 16 sub-blocks, transformed by a two-dimensional discrete cosine transform (DCT), (Equation 6.3), and categorized into one of four activity classes depending on the total ac energy within the 16 x 16, 8 x 8 (zonal) or 4 x 4 (zonal) sub-blocks as shown in Figure 4.9. The activity index within a particular frame is determined using Equations 4.15-4.19 in Chapter IV. The 24 blocks within each frame are then reordered with respect to their activity, and categorized into four classes each of which consist of 6 blocks. The block variances, the standard deviation and the bit assignment matrices are calculated and the normalized transform coefficients within each class are nonuniformly quantized and coded using Max's optimal quantization scheme [69] discussed in Section 4.3.4.1. The quantizer output is coded in binary (variable word length) form and passed to the buffer. At the receiver the binary data is first stored in the receiver buffer, then decoded and inverse transformed for display.

As mentioned in Chapter IV, bits are assigned by initializing the distortion parameter and iteratively calculating the summation of the number of bits in Equation 4.23 until the desired total number of bits is achieved for each class [63]. For a given transmission bit rate and subjective picture quality, then, what should the value of the
FIG. 6.1 ORIGINAL HEAD AND SHOULDERS SCENE CONSISTING OF EIGHT FRAMES
distortion parameter be, and how should it change from frame to frame? In other words, how do we control the transmission rate through the control of this parameter for a given data rate over the channel? Five schemes have been investigated, some of them including buffering to smooth the bit rate over the channel. These schemes assign the number of bits for each transform coefficient adaptively according to image statistics using the activity index classification method (see Appendix III).

6.4 EXPERIMENTAL WORK

6.4.1 Bit Assignment And Bit Rate Control In Intraframe And Interframe Image Processing

To optimize adaptive transform coding systems with respect to mean square error, during bit allocation one should assign the same value of distortion parameter to all transform coefficients within all classes, and this result is supported by the rate distortion equation (4.23 in Section 4.3.5.3) for bit allocation.

However, it may be possible to enhance the subjective performance of this type of system by having different values of distortion parameter for each class, perhaps allowing a larger distortion in the higher activity classes for certain values of transform coefficient variance, since in the higher activity class a given $\sigma^2(u,v)$ in Equation (4.21) will correspond to a higher spatial frequency than
\( \sigma_2^2 (x,y) \) of the same magnitude in, say, the second activity class. The argument here is that the frequency of \( \sigma_1^2(u,v) \) is perhaps subjectively less visible, and can tolerate higher mean square error.

6.4.1.1 Scheme I

This scheme involves the bit assignment for four classes in each frame and for eight frames with distortion parameter (Equation 4.23) constant and equal to 5. i.e. for the frame of 24 blocks, each of which has \((16 \times 16)\) transform coefficients, four classes are assigned using the activity index strategy which results in a classification map and four bit allocation matrices for each frame. The classification map and bit allocation matrices for the first frame are shown in Table 6.1. The activity index assignment in this scheme has been carried out on the basis discussed in Chapter IV Section 4.3.5.2 for a \(16 \times 16\) coefficient block and \(8 \times 8\) and \(4 \times 4\) element zonal sub-blocks as a first step for the purpose of examining the performance of the classifier for moving image sequences. A comparison between the activity index for each block within each frame and throughout eight frames for \(16 \times 16\) blocks, and \(8 \times 8\), and \(4 \times 4\) zonal sub-blocks has been carried out and, again, shows validity of the use of the \(4 \times 4\) zonal sub-block for classification in order to reduce the number of computational operations. The \(4 \times 4\) zonal sub-block has thus been used for this scheme and for all other schemes discussed in this chapter for class assignment. The results for the conventional scheme
Table 6.1  (a) The Classification Map For The Frame Of 
(4 x 6) Blocks

(b),(c),(d),(e) Bit Allocation Matrices For 
The Four Classes Within The 
First Frame.
are given in Table 6.II. In the table, the number of the frames, the total number of bits within each activity class, the average bit rate for each frame, and the normalized mean square error for each frame are shown. From the table it should be noted that the total number of bits within class two of the fifth frame is higher than the total number of bits within the first class in the same frame. The explanation for this phenomenon may be that each frame consists of only 24 blocks, giving 6 blocks at the output of the classifier (Figure 4.9, Chapter IV) within each class, and that, when the number of blocks in each class is small, the curve in Figure 4.18 (Chapter IV) may not be applicable, for two possible reasons. First, the number of variance averages is too small within each class (6 in this case). Secondly, the index method of classification, which depends on the summation of the a.c. signal power in each block has disadvantages for small size image processing, because if there were two different transformed blocks as shown in Figure 6.2, the variance

FIG 6.2 REPRESENTS TWO TRANSFORMED BLOCKS WITH DIFFERENT ACTIVITY
(a) HIGHLY DETAILED BLOCK
(b) HALF GREY AND HALF WHITE BLOCK
<table>
<thead>
<tr>
<th>Number of frame</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1</td>
<td>class 2</td>
<td>class 3</td>
</tr>
<tr>
<td>1</td>
<td>344</td>
<td>308</td>
<td>222</td>
</tr>
<tr>
<td>2</td>
<td>372</td>
<td>261</td>
<td>190</td>
</tr>
<tr>
<td>3</td>
<td>385</td>
<td>318</td>
<td>285</td>
</tr>
<tr>
<td>4</td>
<td>309</td>
<td>256</td>
<td>243</td>
</tr>
<tr>
<td>5</td>
<td>382</td>
<td>389</td>
<td>312</td>
</tr>
<tr>
<td>6</td>
<td>309</td>
<td>309</td>
<td>273</td>
</tr>
<tr>
<td>7</td>
<td>413</td>
<td>389</td>
<td>315</td>
</tr>
<tr>
<td>8</td>
<td>332</td>
<td>299</td>
<td>273</td>
</tr>
<tr>
<td>Average</td>
<td>355</td>
<td>310</td>
<td>264</td>
</tr>
</tbody>
</table>

Table 6.II  Scheme I with D = 5.

Overall rate = 1.08 bits/element
of the highly detailed block "a" could be less than the variance of block "b" which is, say, half grey and half white. In this case block "b" will have the higher activity level (say class one) and block "a" the lower activity. However, it is obvious that the total number of bits needed to represent block "a" will be more than that needed to represent block "b", and this might result in the total number of bits in the second class (lower activity than the first) being greater than the total number of bits in the first, under the condition of constant distortion parameter for all classes within the frame and for all frames. This will affect transmission over the channel if classification map control is used to prevent buffer overflow and underflow, and therefore other schemes are introduced here for bit rate control. In some of them the problems of bit distribution among the classes of classified frames are considered. Figure 6.3(b) shows eight processed frames with overall bit rate equal to 1.08 bits per element. For further bit rate reduction the distortion parameter is increased to a value of 30 giving a rate of 0.5 bits per element with good picture quality as shown in Figure 6.4 and Table 6.11.

6.4.1.2 Scheme II

In this scheme the bit assignment is carried out in a way similar to that of Chapter IV, (Section 4.4, Table (4.4), where the desired number of bits for the processed
FIG. 6.3 RESULTS FOR SCHEME I
(a) ORIGINAL PICTURES
(b) PROCESSED PICTURES
THE NUMBERS (1-8) INDICATE THE TEMPORAL SCANNING DIRECTION
<table>
<thead>
<tr>
<th>Number of frames</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1</td>
<td>class 2</td>
<td>class 3</td>
</tr>
<tr>
<td>1</td>
<td>162</td>
<td>160</td>
<td>109</td>
</tr>
<tr>
<td>2</td>
<td>159</td>
<td>140</td>
<td>105</td>
</tr>
<tr>
<td>3</td>
<td>190</td>
<td>137</td>
<td>134</td>
</tr>
<tr>
<td>4</td>
<td>172</td>
<td>128</td>
<td>126</td>
</tr>
<tr>
<td>5</td>
<td>185</td>
<td>189</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>168</td>
<td>159</td>
<td>136</td>
</tr>
<tr>
<td>7</td>
<td>193</td>
<td>191</td>
<td>144</td>
</tr>
<tr>
<td>8</td>
<td>177</td>
<td>153</td>
<td>136</td>
</tr>
<tr>
<td>Average</td>
<td>175</td>
<td>155</td>
<td>130</td>
</tr>
</tbody>
</table>

Table 6.III  \( D = 30 \) for all frames.

Overall rate = 0.5 bits/element
FIG. 6.4  RESULTS FOR SCHEME I WITH OVERALL RATE = 0.5 BIT PER ELEMENT
(a) ORIGINAL PICTURE
(b) PROCESSED PICTURE
image is iteratively achieved for a given bit rate, or desired image quality. Here, also, different values of the distortion parameter are assigned for different classes. For the four classes D is equal to 5, 4, 4 and 6 respectively, allowing a comparison between bit assignment for still pictures, namely the "Girl", "Test card" and "Flat", and interframe bit assignments to the frame sequence. The results of this scheme are shown in Table (6.IV) and Figure 6.5. In Table (6.IV) the total number of bits for each class, the average bit rate, and the normalized mean square error are shown for each frame, and the overall average bit rate of 1.12 bits per element is achieved with intraframe transform coding. This scheme is even worse than Scheme I in terms of the distribution of the total number of bits between classes as can be seen from Table (6.IV) for frames 5, 6, and 7 for the reasons mentioned in connection with the previous scheme (Figure 6.2). In the next three schemes therefore, the bit rate is controlled and the distribution of bits is corrected.

6.4.1.3 Scheme III

This scheme allows the distortion parameter to be controlled according to the total number of bits within each class. The value of distortion parameter (D) was initialised at 5 for the first class, considering the maximum total number of bits within the first class as an approximate indicator of a given bit rate. Once the bit
<table>
<thead>
<tr>
<th>Number of frames</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1</td>
<td>class 2</td>
<td>class 3</td>
</tr>
<tr>
<td>1</td>
<td>344</td>
<td>335</td>
<td>244</td>
</tr>
<tr>
<td>2</td>
<td>372</td>
<td>274</td>
<td>209</td>
</tr>
<tr>
<td>3</td>
<td>385</td>
<td>344</td>
<td>306</td>
</tr>
<tr>
<td>4</td>
<td>309</td>
<td>277</td>
<td>263</td>
</tr>
<tr>
<td>5</td>
<td>382</td>
<td>429</td>
<td>339</td>
</tr>
<tr>
<td>6</td>
<td>309</td>
<td>337</td>
<td>293</td>
</tr>
<tr>
<td>7</td>
<td>413</td>
<td>426</td>
<td>341</td>
</tr>
<tr>
<td>8</td>
<td>332</td>
<td>319</td>
<td>301</td>
</tr>
<tr>
<td>Average</td>
<td>355</td>
<td>342</td>
<td>287</td>
</tr>
</tbody>
</table>

Table 6.IV  Scheme II with a different value of D for each class. Overall rate = 1.12 bits/element
FIG. 6.5 RESULTS FOR SCHEME II
(a) ORIGINAL PICTURES
(b) PROCESSED PICTURES
allocation matrix $N_{b1}(u,v)$, for the first class has been computed, and the total number of bits, say $N_{b1}$ is known, the other bit allocations are determined as follows:

The bit allocation matrix for the first class has been computed with $D = \text{constant}$ (in this case equal to 5). To calculate the bit allocation matrix for the second class a new technique has been developed in which the bit rate is controlled and the distribution of the total number of bits in each class corrected by setting the distortion parameter adaptively according to the instantaneous total number of bits within the second class (say $N_{b2}$) during the computation of the bit allocation matrix of this class. To do this a comparison between the instantaneous sum of the number of bits in the present class and the instantaneous sum of the number of bits in the previous class is made. Since the ac coefficient energy is compacted in the upper left corner of the variance matrix the following equation is used to estimate the distortion parameter for the $(i,j)$th location:

$$C_I = \sqrt{i^2 + j^2}$$

(6.7)

where $C_I$ is taken to be the radial spatial frequency.

The instantaneous total number of bits for a particular class is given by

$$N_{bk} = \sum_{i=1}^{N} \sum_{j=1}^{N} \text{BIT} (i,j)$$

(6.8)
where \( i = 1, 2, \ldots, N, j = 1, 2, \ldots, N, k = 1, 2, 3, 4, \)
and \( k \) is the number of the class.

To calculate the bit allocation matrix for the second class the distortion parameter is initialized for the first location of the block variance matrix \( \sigma_k^2(u,v) \), which corresponds to \( \sigma_k^2(1,1) \) or BIT \((1,1)\) in the bit allocation matrix, and Equation (6.8) is used, where \( k > 1 \).

From the first calculation step, we will have the instantaneous number of bits for the second class say \( N_{b2}(i,j) \) where \( i=1 \) and \( j=1 \), and we can compare \( N_{b2}(i,j) \) with \( N_{b1}(i,j) \). \( WF \), where \( WF \) is constant, and \( N_{b1}(i,j) \) is the instantaneous number of bits for the first class. In the next step of the calculation, if \( N_{b2}(i,j) > N_{b1}(i,j) \).\( WF \) the D-distortion parameter is changed in Equation [(4.23) in Chapter IV] using

\[
D = D_1 \left[1 + \left(\frac{CI}{100}\right)\right] \left(\frac{N_{b2}}{N_{b1}} \cdot WF\right)
\]

(6.9)

where \( D_1 \) is equal to a constant and \( CI \) has been derived from Equation (6.7). The weighting factor \( WF \) is equal to 0.90. Other parameters have been chosen experimentally to give the best overall performance in terms of bit rate control and subjective picture quality. Table 6.V and Figure 6.6 present the results of this scheme.

6.4.1.4 Scheme IV

The bit assignment method in this scheme is a combination of those of the first and third schemes, but instead of
<table>
<thead>
<tr>
<th>Number of frames</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1</td>
<td>class 2</td>
<td>class 3</td>
</tr>
<tr>
<td>1</td>
<td>346</td>
<td>294</td>
<td>218</td>
</tr>
<tr>
<td>2</td>
<td>372</td>
<td>248</td>
<td>187</td>
</tr>
<tr>
<td>3</td>
<td>388</td>
<td>314</td>
<td>259</td>
</tr>
<tr>
<td>4</td>
<td>310</td>
<td>254</td>
<td>227</td>
</tr>
<tr>
<td>5</td>
<td>385</td>
<td>359</td>
<td>306</td>
</tr>
<tr>
<td>6</td>
<td>309</td>
<td>287</td>
<td>260</td>
</tr>
<tr>
<td>7</td>
<td>421</td>
<td>365</td>
<td>304</td>
</tr>
<tr>
<td>8</td>
<td>334</td>
<td>291</td>
<td>265</td>
</tr>
</tbody>
</table>

| Average          | 358      | 301      | 253      | 190      | 1.06             | 0.08             |

Table 6.V  Scheme III with D1 = 5. Overall rate = 1.06 bits/pel
FIG. 6.6 RESULTS FOR SCHEME III
(a) ORIGINAL PICTURES
(b) PROCESSED PICTURES
changing the distortion parameter using equation (6.9), the value of D is fixed for all classes and all frames as in Scheme I. The instantaneous total number of bits in the present class, say $N_{bk}^{k}(i,j)$, where $k>1$, is compared with the instantaneous total number of bits within the previous class say, $N_{bk}^{k-1}(i,j)$. If $N_{bk}^{k}(i,j)$ is greater than $N_{bk}^{k-1}(i,j)$.WF, one bit from the same location in the present class is removed. The number of bits $N_{bk}^{k}(i,j)$ is then calculated using Equation (6.8) and the value of D is 5. The factor WF was chosen experimentally and again set equal to 0.9. The operation is given mathematically by

$$N_{bk}^{k} = \sum_{i=1}^{N} \sum_{j=1}^{N} BIT(i,j) - 1 \quad (6.10)$$

for $k>1$ and $BIT(i,j)$ is computed using Equation 4.23 in Chapter IV. Table 6.VI and Figure 6.7 show the results for this scheme.

6.4.1.5 Scheme V

Interframe activity is considered in this scheme where the derived activity index is compared between successive frames for each block. The transform blocks are generated using Equation (6.3) and the activity index for each block is determined by

$$AI = \sigma_{i,j,t}^2 = \frac{1}{R^2} \sum_{u=0}^{R-1} \sum_{v=0}^{R-1} [T_{i,j,t}(u,v)]^2 \quad (6.11)$$

for $(u,v) \neq 0$
<table>
<thead>
<tr>
<th>Number of frames</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1 class 2 class 3 class 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>344 308 222 164</td>
<td>1.0</td>
<td>0.10</td>
</tr>
<tr>
<td>2</td>
<td>372 249 190 152</td>
<td>0.94</td>
<td>0.09</td>
</tr>
<tr>
<td>3</td>
<td>385 318 261 214</td>
<td>1.15</td>
<td>0.11</td>
</tr>
<tr>
<td>4</td>
<td>309 256 227 183</td>
<td>0.95</td>
<td>0.08</td>
</tr>
<tr>
<td>5</td>
<td>382 347 312 215</td>
<td>1.22</td>
<td>0.09</td>
</tr>
<tr>
<td>6</td>
<td>309 287 267 183</td>
<td>1.02</td>
<td>0.07</td>
</tr>
<tr>
<td>7</td>
<td>413 379 315 233</td>
<td>1.30</td>
<td>0.08</td>
</tr>
<tr>
<td>8</td>
<td>332 299 275 189</td>
<td>1.06</td>
<td>0.06</td>
</tr>
<tr>
<td>Average</td>
<td>357 305 258 191</td>
<td>1.08</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6.VI  Scheme IV with D = 5 and overall bit rate = 1.08
FIG. 6.7 RESULTS FOR SCHEME IV
(a) ORIGINAL PICTURES
(b) PROCESSED PICTURES
where \( R^2 = N^2 - k \), and \( k \) is the number of discarded transform coefficients, \( N \times N \) is the dimension of transform block, \( R \times R \) is the dimension of the restricted region chosen to define the activity, \( i = 1,2, \ldots, 4, \ j = 1,2, \ldots, 6 \), represent the location of a particular block, and \( t \) is the frame number. The activity index of a particular block within the previous frame \( a_{i,j,t-1} \), and the activity index of the corresponding block in the present frame are used here to improve the subjective picture quality which results from the use of interframe activity within the bit assignment algorithm, where the number of bits for a particular block is adaptively assigned with respect to that activity, i.e. more bits are assigned to the present block by decreasing the value of distortion parameter if it is more active, and the value of \( D \) is given by

\[
D = D_1 \left( \frac{\sigma^2_{i,j,t-1}}{\sigma^2_{i,j,t}} \right) \quad (6.12)
\]

where \( D_1 \) is the distortion parameter for the first frame, and the bit assignment is carried out using Scheme IV. It should be pointed out that this scheme is an extension of Scheme IV involving consideration of interframe activity. The process is illustrated in Figure 6.8 and the results obtained are presented in Table 6.VII and Figure 6.9.

In this scheme, as well as in the others, both normalized mean square error (NMSE) and subjective quality of the processed images have been considered. The NMSE for each block throughout each frame and for eight processed frames
\( \sigma^2_{1,1,t-1} \) is the variance of the first block within the previous frame.

\( \sigma^2_{1,1,t} \) is the variance of the first block within the present frame.

FIG. 6.8 INTERFRAME BLOCK VARIANCE COMPARISON.
<table>
<thead>
<tr>
<th>Number of frames</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1</td>
<td>class 2</td>
<td>class 3</td>
</tr>
<tr>
<td>1</td>
<td>344</td>
<td>308</td>
<td>222</td>
</tr>
<tr>
<td>2</td>
<td>378</td>
<td>261</td>
<td>192</td>
</tr>
<tr>
<td>3</td>
<td>385</td>
<td>222</td>
<td>205</td>
</tr>
<tr>
<td>4</td>
<td>320</td>
<td>270</td>
<td>254</td>
</tr>
<tr>
<td>5</td>
<td>395</td>
<td>390</td>
<td>310</td>
</tr>
<tr>
<td>6</td>
<td>305</td>
<td>304</td>
<td>270</td>
</tr>
<tr>
<td>7</td>
<td>398</td>
<td>307</td>
<td>242</td>
</tr>
<tr>
<td>8</td>
<td>324</td>
<td>299</td>
<td>271</td>
</tr>
<tr>
<td>Average</td>
<td>356</td>
<td>295</td>
<td>246</td>
</tr>
</tbody>
</table>

Table 6.VII  Scheme V with D1 = 5 and overall rate = 1.04 bits/sample
FIG. 6.9 RESULTS FOR SCHEME V
(a) ORIGINAL PICTURES
(b) PROCESSED PICTURES
has been computed and plotted to assess the results for all the schemes investigated. For example, Figure 6.10 shows the NMSE of this scheme for 1-8 frames, where each plot represents the NMSE/number of blocks relation within each frame. It should be pointed out that for blocks with very low levels of luminance the block NMSE measure of noise performance becomes very large as can be seen from Figure 6.10 for blocks number one, seven, and thirteen in frames one and two. In the limiting case the energy of such blocks tends towards zero, and therefore the NMSE becomes very high.

In addition, Figure 6.11 shows the original source data, the reconstructed data, and the difference between the original and reconstructed data respectively for block number one in frame eight.

6.5 BUFFERING TECHNIQUES FOR MOVING IMAGES

Buffering techniques are now incorporated into schemes III, IV, and V which were discussed in Section 6.4. Classification map control according to buffer status to prevent buffer underflow and overflow has been used with Scheme IVb discussed in Chapter V, except that here each frame of (64 x 96) elements instead of (256 x 256) elements is transformed, classified, adaptively quantized, binary coded and passed to the variable input-fixed output bit rate buffer. The receiver processing inversely duplicates that of the
FIG. 6.10  NMSE VERSUS NUMBER OF BLOCK WITHIN FRAME 1 - 8

NMSE
FIG. 6.11 ILLUSTRATING FOR A TYPICAL PROCESSED BLOCK

(a) THE ORIGINAL DATA
(b) THE PROCESSED DATA
(c) THE DIFFERENCE BETWEEN THE ORIGINAL AND PROCESSED DATA
MAGNITUDE

ELEMENT NUMBER

WITHIN A (16x16) BLOCK
transmitter, and buffers in both the transmitter and receiver are larger than for still images due to the transmission of overhead information for each frame. The overhead information which must be transmitted is the classification map and the bit allocation matrices. Two bits are assigned for the classification map (i.e. for four classes), and three bits are assigned to transmit the entries within the four bit allocation matrices (see Table 6.I). The dynamic range of the controller is set equal to 256, 3500, 4000 bits which represent $Th_3$, $Th_1$, $Th_2$, respectively. The pattern insertion technique, block delay and classification map control are included to prevent buffer underflow and overflow. Two blocks are delayed at the start of transmission, i.e., before the first frame is transmitted. The frequency of insertion is low here throughout all frames since threshold $Th_3$ in the controller forces class $i$ to be changed to class $i-1$, in order to prevent buffer underflow. Buffer behaviour for eight transmitted frames within Scheme III (Section 6.4.3) is shown in Figure 6.12. From the figure the maximum buffer size needed is about 3500 bits with an average transmission bit rate equal to 1.5 bits per element, including the transmission of overhead information. The amount of the delay in the transmitter buffer is shown at the beginning of the graph. At the receiver the same number of blocks must be delayed. Also, the presence of overhead information causes an increase in the time delay within the adaptive communication system. Therefore procedures have been considered to reduce the amount of overhead
FIG. 6.12 SIMULATED BUFFER BEHAVIOUR FOR EIGHT TRANSMITTED FRAMES WITH DELAY, INSERTION, AND CLASSIFICATION MAP CONTROL WITHIN SCHEME III OF SECTION 6.4.3.
information needed which will be discussed in the next chapter. The buffer behaviour in Schemes IV and V [Sections 6.4.4 and 6.4.5] is similar to that in Figure 6.12 since the average bit rate for Schemes III, IV, and V are very similar (see Tables 6.V, 6.VI, and 6.VII), and the average transmission rate is the same for all schemes. The processed and transmitted pictures are shown in Figures 6.6, 6.7, and 6.9 respectively.

6.6 DISCUSSION

In adaptive transform image coding of still pictures [Chapter IV] the desired average bit rate is usually achieved by iteratively adjusting the distortion parameter. In the coding of moving images alternative methods can be employed. Five schemes are investigated here to control the bit rate over the channel. Classification map control to prevent buffer underflow and overflow is also incorporated within Schemes III, IV, and V. In Scheme I the same value of distortion parameter is used for all classes and frames. A similar fixed distortion parameter approach is used in Scheme II, but here each class is assigned a different constant value. The results for Scheme I with an average bit rate equal to 0.5 - 1 bit per element are shown in Figure 6.3(b), 6.4, and Tables 6.II, 6.III. The results of Scheme II are shown in Figure 6.5 and Table 6.IV. These two schemes are conventional with regard to bit assignment, and appear inaccurate where the distribution of the number of bits between classes is concerned, perhaps for the reasons
discussed in Section 6.4.1.1, (see Figure 6.2). In Schemes III, IV, and V an alternative method is used to distribute bits, to control the desired average bit rate and to improve subjective picture quality. In Scheme III, the value of the distortion parameter is initialised for the most active class, considering the maximum total number of bits within this class as an approximate indicator of a given bit rate. The distribution of the total number of bits in each class is then corrected by setting the distortion parameter adaptively according to the instantaneous total number of bits within the second class. This operation is then repeated for the third and fourth classes. Good results are achieved and the performance of this scheme is demonstrated in Figure 6.6 and Table 6.V with average rate equal to 1.06 bits per element. From the comparison of the results of this scheme in Figure 6.6 and those of Scheme I [Figure 6.3(b)] and II (Figure 6.5), and the original pictures (Figure 6.3(a), it can be seen that good picture quality is achieved with little image degradation. Also, a good result is achieved for the control of the distribution of the number of bits between classes and the average rate over the channel. Scheme IV is a combination of the first and third schemes. From the results shown in Figure 6.7 and Table 6.VI, it can be seen that good picture quality is achieved with an average rate over eight frames equal to 1.08 bits per element and average normalized mean square error (NMSE) equal to 0.08%. In
addition this scheme is more simple than Scheme III in terms of the number of computational operations, but, unlike Scheme V it does not consider interframe activity. In Scheme V the activity of each block in the present frame is compared with the corresponding activity index in the previous frame. The distortion parameter is computed for the present block as a function of this comparison and the initialized value. A good result is achieved with lower average bit rate and the same average NMSE of Scheme III as can be seen from Figure 6.9 and Table 6.VII. From the overall figures for bit rate, NMSE, bit distribution between classes and subjective picture quality of the processed frames Scheme V does appear to be superior to other schemes, as might be expected, since it takes into account interframe activity to change the distortion parameter adaptively. Buffering techniques with classification map control according to buffer status have been incorporated within Schemes III, IV, and V. The buffer behaviour is shown in Figure 6.12 where the maximum buffer size is about 3500 bits for Scheme III. Since the average bit rate for all schemes is about 1 bit per element, the buffer behaviour for Schemes IV, and V will be very similar to that shown in Figure 6.12.

The overall delay in any communication system caused by buffering is equal to the transmitter and receiver buffer sizes and must satisfy the equation [31]

\[
\frac{(B_{TX} + B_{RX})}{R} + T \leq 600 \text{ m sec} \tag{6.13}
\]
where $B_{TX}$ is the transmitter buffer size, $B_{RX}$ is the receiver buffer size, $R$ is the transmission channel rate (bits/second), and $T$ is the total round-trip channel delay, to avoid annoying delay effects. In the present case, if a memory store RAM of size 8k bits is suggested with average bit rate $br$ 1 bit per element, $R$ will be equal to $1 \times 256 \times 256 \times 25$ bits per second for an image of 256 x 256 elements. Then $(B_{TX} + B_{RX}) / R$ is equal to 

$$(2 \times 8 \times 1024 \text{ bits}) / (1 \times 256 \times 256 \times 25 \text{ bits/sec}) = 10 \text{ ms}$$

and it is evident that the overall system delay caused by the buffering technique used here is less than 10 m.sec. since the buffer size is not more than 3500 bits.

6.7 CONCLUSIONS

For a given transmission rate, the designer of the adaptive communication system should take into account the variable rate of the encoder output as well as system complexity. System complexity in terms of computational operation reduction has been considered in Chapter IV for still images, and in the present chapter for moving images. It is found that adaptive image coding using activity index classification can be applied to the coding of moving images with a large degree of computational reduction if 4 x 4 zonal sub-blocks are used for activity index evaluation. Also, it is concluded that the distribution of the total number of bits between classes and the overall average rate
can be controlled within the intraframe coding algorithm before passing the information bit stream to the variable bit rate buffer.

In Schemes I and II the comparison between bit assignment for still pictures and moving images has been made for activity index coders, and it has been shown that the iterative method of establishing the desired rate can be modified by using Scheme III, IV, or V to achieve approximate rate control over the channel. However, Schemes III and IV have been used to control the bit rate within intraframe activity coder. In Scheme V, interframe and intraframe activity have been considered where the distortion parameter D is changed according to interframe activity (Equation 6.12) and this scheme performs better than others in terms of subjective picture quality (as can be seen from Figure 6.9). Furthermore, variable bit rate buffering for moving images has been investigated using classification map control according to buffer status to prevent buffer underflow and overflow with the maximum buffer size of 3500 bits including overhead transmission through the buffer. Therefore an 8 k bit RAM memory store is suggested for use as a buffer in systems for adaptive transform coding of moving images without annoying delay effects in communication.
CHAPTER VII

INTRA-FRAME AND INTER-FRAME ENERGY ESTIMATION
Minimisation of processing complexity and overhead requirement are important considerations for the system designer. In this chapter two approaches to this problem are described. In the first, a new method of interframe energy estimation using the classification map within an adaptive transform coding system (see Chapters IV, V and VI) is introduced to reduce the number of computational operations and the overhead load. In the second scheme, the overhead requirement and also the buffer size have been reduced by estimating the variance of each transform coefficient at the receiver. Both schemes achieve good results in terms of subjective picture quality.

It has been indicated in Chapter VI (Sections 6.1 and 6.2) that various techniques have been used by workers in the field for interframe coding, such as interframe DPCM [40], conditional replenishment [42], and motion compensation [43, 88, 98]. Most of the previous techniques involve element-to-element coding between frames to reduce the transmission bit rate, and so storage of two frames is needed at the transmitter and receiver, irrespective of algorithm complexity. Instead of using the above-mentioned techniques, here the classification maps have been used for interframe coding.

In Section 7.2 the investigation of interframe adaptive transform coding using the classification map will be discussed, and Section 7.3 will describe intraframe energy estimation at the receiver. Section 7.4 contains the discussion and conclusions.
7.2 INTERFRAME ENERGY ESTIMATION SCHEME USING CLASSIFICATION MAPS (SCHEME I)

Figure 7.1 shows the block diagram of an adaptive transform coder with interframe energy estimation using frame classification maps. The input signal is transformed using the two-dimensional cosine transform (DCT) and sub-blocks allocated to one of four classes. The classification map and bit allocation matrices are adaptively quantized, coded and passed to the buffer. This is carried out for the first and the second frame. Having the classification maps, say CM(i,j,k) and CM(i,j,k-1) for the first and second frames at the (i,j)th block location and frame number k, the block activity change from frame to frame can be determined. Figure 7.2 shows the classification maps of three successive frames. The determination of block activity change from class to class, say class L to L-1 for L>1 or L to L+1 for L<4 where L = 1,2,3,4, is carried out by the comparison between classification maps. For example, from Figure 7.2, if CM(i,j,k-2) and CM(i,j,k-1) are compared and the result of this comparison is greater than or equal to a predetermined threshold Th, the estimation, class assignment, and bit allocation matrices for frame "k" are determined and the switch "k" in Figure 7.1 is left in "normal" position. If the comparison is less than Th, frame k uses the same classification map and bit allocation matrices as for the frame (k-1)th. In this case a signal from the comparator alters switch "k" to the "skip" position.
FIG. 7.1 BLOCK DIAGRAM OF AN ADAPTIVE TRANSFORM CODER WITH AN INTER-FRAME ENERGY ESTIMATION SCHEME USING FRAME CLASSIFICATION MAPS
FIG. 7.2 ILLUSTRATING CLASSIFICATION MAPS OF THREE SUCCESSIVE FRAMES

- 332 -

Classification map of frame K-2

Classification map of frame K-1

Classification map of frame K

x - Denotes block activity transition between the K-2, K-1, and Kth frames respectively.
In this experiment the threshold $Th_c$ is set equal to 8, allowing two frames out of eight to be skipped. The results of eight processed frames with and without skipping are shown in Figure 7.3. This technique results in a reduction of up to one third in processing time and the number of computational operations with a good performance in terms of subjective picture quality and normalized mean square error (see Figure 7.3 and Table 7.1). The value $Th_c$ can be set equal to 2, 4, and 8. For the value of 2 there is no skipping and estimation of each frame is carried out. For $Th_c=4$ only the third frame was skipped, and for a value of 8 the third and the seventh frames are skipped, which means that by adjusting ($Th_c$) the overall system delay caused by the classification and bit allocation procedures can be varied. In real time systems however, if a frame is skipped and not transmitted, the previous frame must be repeated at the receiver to prevent flicker effect. In this case a frame store at the receiver is needed and a flag signal is sent to inform the receiver of the need for frame repetition. It should be pointed out that the value $Th_c=8$ is for a total of 24 blocks. For an image consisting of 256 blocks (each of which has $(16 \times 16)$ elements) the value of $Th_c$ may be increased.

7.3 INTRAFRAME ENERGY ESTIMATION AT THE RECEIVER

(SCHEME II)

In this scheme the intraframe energy is estimated at the receiver within the adaptive transform coding system.
<table>
<thead>
<tr>
<th>Number of frame</th>
<th>Number of bits within each class</th>
<th>Average bit rate</th>
<th>NMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>class 1</td>
<td>class 2</td>
<td>class 3</td>
</tr>
<tr>
<td>1</td>
<td>344</td>
<td>308</td>
<td>222</td>
</tr>
<tr>
<td>2</td>
<td>378</td>
<td>261</td>
<td>192</td>
</tr>
<tr>
<td>3 skipped</td>
<td>378</td>
<td>261</td>
<td>197</td>
</tr>
<tr>
<td>3 unskipped</td>
<td>385</td>
<td>222</td>
<td>205</td>
</tr>
<tr>
<td>4</td>
<td>320</td>
<td>270</td>
<td>254</td>
</tr>
<tr>
<td>5</td>
<td>395</td>
<td>390</td>
<td>346</td>
</tr>
<tr>
<td>6</td>
<td>305</td>
<td>304</td>
<td>270</td>
</tr>
<tr>
<td>7 skipped</td>
<td>305</td>
<td>304</td>
<td>270</td>
</tr>
<tr>
<td>7 unskipped</td>
<td>398</td>
<td>307</td>
<td>242</td>
</tr>
<tr>
<td>8</td>
<td>324</td>
<td>299</td>
<td>271</td>
</tr>
<tr>
<td>Averages</td>
<td>343.6</td>
<td>299.6</td>
<td>247.6</td>
</tr>
</tbody>
</table>

Table 7.I  Results for the interframe energy estimation scheme using classification maps.
Overall rate = 1.04 bits/element.
Overall NMSE = 0.09%
(Frames 3 and 7 are skipped).
FIG. 7.3 RESULTS FOR INTER-FRAME ENERGY ESTIMATION USING FRAME CLASSIFICATION MAPS.
(a) PROCESSED PICTURES WITHOUT SKIPPING
(b) PROCESSED PICTURES WITH SKIPPING
The block diagram of the system used is shown in Figure 7.4(a,b). The adaptive coding procedure eliminates the need for classification and the algorithm utilizes a (16 x 16) transform block size for the (64 x 96) moving image. The normalization coefficients and bit assignment are recursively derived on the basis of previously decoded information. To do this the variance of each transform coefficient is dynamically estimated along the scan path shown in Figure 7.5(a). It has been shown [91,133] that the logarithm of the variances of the transform coefficients are highly correlated with the coefficient index along this zig-zag path. This process maps a two-dimensional transform block $T(u,v)$ into one dimension $T(k)$. Since the sub-diagonal elements have roughly the same relative importance, the mapping results in an approximately monotonically decreasing set of values. If the one-dimensional sequence of transform coefficients $T(k)$ is coded let the corresponding decoded sequence be $\hat{T}(k)$. The dynamically estimated variance represented by $\hat{\sigma}^2(k)$ is given by

$$\hat{\sigma}^2(k) = W\hat{\sigma}^2(k-1) + (1-W) \hat{T}'(k-1)$$ (7.1)

where the weighting factor $W$ was chosen experimentally to give the best estimation [62,133]. The value $\hat{\sigma}^2(3)$ is computed from adjacent previous transmitted decoded values of $\hat{T}'(k)$

$$\hat{\sigma}^2(3) = \frac{1}{2} \sum_{k=1}^{N} \hat{T}^2(k)$$ (7.2)
FIG. 7.4(a) INTRA-FRAME VARIANCE ESTIMATION ENCODER.
FIG. 7.4(b) INTRA-FRAME VARIANCE ESTIMATION DECODER.
where \( N = 2 \). In order to initialize the variance estimation at the receiver, the transform coefficients \( T(k), \{k = 1, 2\} \) for all image blocks are transmitted with 7 bits. Having estimated \( \hat{\sigma}^2(k) \), the bit assignment is then carried out using Equation 4.23. For the transformed block of 256 coefficients variance estimation is stopped when the number of bits assigned to the Kth coefficients becomes less than one. A typical bit allocation matrix is shown in Figure 7.5(b). The assigned number of bits, say \( IB(k) \), is needed for deriving the variable code word from the buffer, binary decoder, reconstruction and determination of the normalization coefficient for the kth location, where the normalization coefficients are given by

\[
\hat{\sigma}'(k) = \frac{IB(k) - 1}{2}
\]  
\[\text{(7.3)}\]

and the inverse normalization is

\[
\hat{T}'(k) = [\hat{T}(k)] [\hat{\sigma}'(k)]
\]  
\[\text{(7.4)}\]

where \( \hat{T}(k) \) is the quantizer output. Inverse two-dimensional transformation using the discrete cosine transform (IDCT) is then applied to generate the eight processed frames shown in Figure 7.6. It should be noted that the coder carries out all the operations mentioned above except for inverse normalization. The quantization strategy used is that described in Chapter IV.

The performance of the simulated system in terms of normalized mean square error per frame and overall average
FIG. 7.5(a) ZIGZAG SCAN PATH USED IN INTRAFRAME VARIANCE ESTIMATION
<table>
<thead>
<tr>
<th>8</th>
<th>7</th>
<th>7</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>3</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**FIG. 7.5(b) TYPICAL BIT ALLOCATION MATRIX**
FIG. 7.6 RESULTS OF SCHEME II
(a) ORIGINAL PICTURE
(b) PROCESSED PICTURE WITH OVERALL AVERAGE
RATE 1.2 BITS/ELEMENT
FIG. 7.7 NMSE VERSUS THE NUMBER OF BLOCKS WITHIN FRAME 1 - 8
Fig. 7.8 Buffer behaviour for scheme II with average transmission rate 1.3 bits/elements and threshold control 180 bits
bit rate equal 1.2 bits/element has been computed. The NMSE for each block within the frame is plotted against the number of the block index as shown in Figure 7.7. The NMSE for each frame is included in Table 7.II. From the comparison of Figure 7.7 and 6.10 in Chapter VI Section 6.4.1.5, it can be seen that the values of NMSE for the processed images here are higher than those of Figure 6.10 (see also Tables 7.II and 6.VII). This demonstrates, in general, the efficiency of adaptive coding techniques using activity indices. However, in the latter overhead load is needed whilst here the overhead information is eliminated and system complexity is reduced.

Buffering has been incorporated within this scheme using the coarse quantization method discussed in Chapter V (Section 5.3.4.5) with the difference that here the controller is operating on the number of bits for the kth coefficient location rather than the bit allocation matrix as a whole, i.e. when the content of the buffer approaches the predetermined threshold (Th) for T (k-1) the assigned number of bits for coefficient T(k) is decreased by one and this requires two flag signals to be sent to inform the decoder of the beginning and the end of the control operation. Buffer behaviour is shown in Figure 7.8 with the average transmission rate equal to 1.3 bits/element. This technique does not need any overhead information and the buffer size is smaller than that of an adaptive transform coding system using an activity index. This scheme is not fully adaptive however, but by using such a technique the overhead information,
<table>
<thead>
<tr>
<th>Number of frame</th>
<th>Normalized mean square (NMSE) %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.19</td>
</tr>
<tr>
<td>2</td>
<td>0.18</td>
</tr>
<tr>
<td>3</td>
<td>0.12</td>
</tr>
<tr>
<td>4</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.13</td>
</tr>
<tr>
<td>6</td>
<td>0.11</td>
</tr>
<tr>
<td>7</td>
<td>0.13</td>
</tr>
<tr>
<td>8</td>
<td>0.09</td>
</tr>
<tr>
<td>Average</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Table 7.II  Performance of intraframe energy estimation technique (Scheme II) in terms of NMSE % for each frame. Overall average rate 1.2 bits/element.
system complexity, and buffer size can be reduced at the cost of a higher transmission rate and NMSE in this case. The eight processed frames are shown in Figure 7.6.

7.4 DISCUSSION AND CONCLUSIONS

From the results for Scheme I (Figure 7.3) where the classification maps are used for interframe energy estimation within an adaptive transform coding system using an activity index, it can be seen that the subjective picture quality achieved with an average rate equal to 1.04 bits per element and average normalized mean square error equal to 0.09% is reasonably good. In Table 7.1, skipped and unskipped frames are tabulated for a comparison of the average rate and the NMSE. From the table the information in frames two and six is used for the transmission of frames three and seven respectively. It should be noted that although the value of the NMSE for the skipped frames is increased over that for the unskipped ones, they are still visually acceptable as can be seen from Figure 7.3(b). (See frame 3 and 7 counting from the right hand side corner).

It can be concluded that using such a technique the number of computational operations can be reduced to one third of the overall number of operations required for activity index and bit allocation assignment. In addition, a further transmission rate reduction can be achieved if the skipped frames are not transmitted, but repeated at the receiver, when the rate becomes equal to 0.34 bits/element.
In Scheme II, intraframe energy estimation for recursive normalization and quantisation is used. Here the classification operation is removed which results in the elimination of the need for overhead information. From the results obtained (Figure 7.9) it can be seen that good picture quality is achieved with average rate equal to 1.2 bits per element and normalized mean square error 0.13% (Table 7.II). Buffering using coarse quantization can be incorporated within this scheme, the buffer size being equal to 220 bits. Again, it can be concluded that using such a scheme the overhead requirement as well as the buffer size can be significantly reduced.
CHAPTER VIII

RECAPITULATION AND SUGGESTIONS
FOR
FURTHER WORK
8.1 INTRODUCTION

The objective of the research work reported here in this thesis is the investigation of adaptive transform coding - decoding systems for the encoding of monochrome images in order to reduce the high bit rate necessary for the transmission of high quality digital image signals. Low bit rate transmission is preferably achieved using adaptive rather than nonadaptive coding techniques, which, however, are relatively more complex, and therefore any means of reducing system complexity becomes desirable. At the same time adaptive coding systems result in a variable bit rate, and output and input buffers are needed at the transmitter and receiver respectively. By using an efficient adaptive transform coding technique an average rate of between 0.34 and 1 bit per element can be achieved with acceptable image quality. Different approaches have been investigated within the areas mentioned above, where system complexity, buffering techniques and the need for overhead information are considered as real problems to be solved.

Initially, in order to become familiarised with picture coding in general, various coding techniques were examined. Then threshold and zonal sampling within an adaptive transform coding system with activity index classification were considered with the emphasis on reduction in the complexity of such systems for simplicity as well as cost reduction.
Two schemes have been investigated in this area and it is found that about 90% of the number of computational operations can be eliminated.

Since adaptive systems result in a variable rate data stream at the output of the encoder, whilst the digital transmission channel requires a fixed data rate, buffers for the transmitter and receiver were computer simulated and five schemes have been investigated for still images to prevent buffer underflow and overflow. In addition buffering techniques have been examined within adaptive coding systems using an activity index for moving images. For the moving image sequence bit rate control was carried out within the bit assignment algorithm before passing the data information to the output rate buffer. Again five schemes were investigated, where the distribution of the number of bits between different activity classes is controlled.

The use of activity indices results in a number of categories or classes which determine the classification map. This map is used for interframe activity estimation to simplify the system and achieve bit rate reduction. However this technique still needs overhead information, and to eliminate this requirement intraframe energy has been estimated at the receiver using recursive bit assignment, quantization, and inverse normalization techniques. Buffering also has been incorporated into the system. Significant buffer size reduction is achieved within this system compared
with that needed for activity index systems.

8.2 THRESHOLD SAMPLING

Transform image coding has been proved to be an efficient means of image coding [11-17], [38-39], [59-65]. In the basic transform image coding method an image is divided into small blocks and each block undergoes a two-dimensional transformation to produce an equal size array of transform coefficients. Among various transforms investigated the cosine transform (DCT) has emerged as the best candidate, and it has been shown that it is nearly optimum, especially when the element correlation \( \rho > 1 \). The DCT has therefore been used throughout all of the present work.

Threshold sampling was first investigated using the two-dimensional DCT, where the input image is divided into 16 x 16 element blocks. A suitable threshold is determined from a study of image statistics and those coefficients whose magnitudes are greater than the threshold are transmitted. At the receiver, the transmitted coefficients are inverse transformed and the image elements displayed. The performance of this system using the normalized mean square error (NMSE) criterion shows that the number of transmitted coefficients can be substantially reduced if the DCT is used, and this results, of course, in relatively high bit rate reduction [134]. For various threshold levels, the average number of transmitted coefficients within each block, are, for example,
25.6, 11.5 and 4.6 with NMSE of 0.02, 0.09, and 0.25% respectively, out of a total of 256 coefficients for the processed "Girl" picture.

8.3 ZONAL SAMPLING

A fixed zonal sampling technique in the transform domain has also been investigated using the DCT. Low order coefficients are chosen, and two schemes are examined, an (8 x 8) zonal sub-block in the first case and a (4 x 4) zonal sub-block in the second. In both schemes the zone of transform coefficients is transmitted and those coefficients outside it are discarded. At the receiver the received coefficients are inverse transformed and the reconstructed picture displayed. Good results are achieved in the first scheme for the processed "Girl" picture with NMSE of 0.08%. For the (4 x 4) zonal scheme blocking effects do appear, however, at an NMSE value of 0.25%.

8.4 ACTIVITY INDEX CLASSIFICATION VIA A SIMPLIFIED ENERGY ESTIMATION METHOD

An adaptive transform coding - decoding system utilising activity index classification has been investigated. Generally, transform image coding performs well on most natural scenes, and a coding rate of about 1.5 bits per element is achievable with no apparent visual degradation. To achieve lower rates with acceptable picture quality, it
is necessary adaptively to quantize transform coefficients so that those blocks of coefficients containing large amounts of energy are allocated more quantization levels and code bits than low energy blocks. Image blocks should therefore be categorised within two or more groups or classes. To do this the activity of each block of say (16 x 16) elements must be computed. In almost all adaptive transform coding designs reported to date, the transform is computed, and transform energy is measured or estimated over all block coefficients on a first pass through the image. The information resulting from the first pass (the classification map and bit allocation matrices), is then utilized to determine the quantization levels and code words for the second pass. This system has the advantage that compression factors can be reduced by a factor of two or more as compared with nonadaptive coding systems. The practical difficulties are the number of computational operations for the first pass and the memory required for the first and second passes. Two different methods of simplifying the activity assignment were examined. Such simplification is brought about by the design of a new classifier to reduce the number of calculations in which, instead of using all coefficients to determine the variance or the sum of the absolute values of the coefficients to classify the image block, only a zonal sub-block of 8 x 8 or 4 x 4 coefficients has been used. Having the activity index for each block, the reordering operation is carried out, and the reordered activity range is divided into equally populated classes. Bit allocation
matrices are derived from the block variance matrices for each class, and the transform coefficients, normalized (equations 4.24 - 25), and quantized employing a $\hat{M}_x$ quantizer [69] with a minimum mean square error criterion. The probability distribution is assumed to be Rayleigh for the DC and Laplacian for the AC coefficients. The quantizer output is binary coded and passed to the variable input-output rate buffer.

Given the satisfactory performance of the simulated activity index system shown in Figure 4.22-30 for the conventional (16 x 16) and for both new approaches using (8 x 8, 4 x 4) zonal sub-blocks and using AC energy or the absolute magnitude of transform coefficients for classification, a reduction in computational load of about 75% and 95% is achieved for the 8 x 8 and 4 x 4 cases respectively. The simulation results show acceptable picture quality in both cases, given that the aim of the investigation is to reduce equipment complexity and cost.

8.5 BUFFERING FOR VARIABLE BIT RATE SYSTEMS

Adaptive coding techniques result in a variable bit rate at the output of the encoder, which changes according to signal statistics. Buffer stores are therefore needed at both the transmitter and receiver when a real-time signal is coded into a varying rate sequence of digits which is to be transmitted over a channel at a uniform rate. Practical stores must be of finite size and therefore may be subject
to overflow and underflow. To prevent these problems, five schemes were investigated, and a new strategy developed. Each frame is divided into sub-blocks which are categorized within four classes according to their activity, and the new strategy is used to control the classification map according to buffer status in order to reduce the likelihood of overflow or underflow.

The simulation of a variable input-fixed output bit rate buffer at the transmitter, and a fixed input-variable output bit rate buffer at the receiver for different source pictures was carried out and relatively small buffers found to be acceptable. In Schemes I and II the underflow problem was studied. To prevent buffer underflow, in Scheme I, a few blocks of data are delayed in the buffer before the transmission is started. In Scheme II a pattern insertion technique was used. It was concluded that to prevent underflow, either data block delay (Scheme I) or pattern-insertion (Scheme II) can be used. Furthermore, a combination of both schemes is also possible. In Schemes IIIa,b and IVa,b a new strategy was applied to prevent buffer underflow and overflow in which the classification map for the image was controlled according to buffer status. In Scheme IIIa,b pattern insertion and classification map control were examined and in Scheme IVa,b pattern-insertion, block delay and classification map control are combined to prevent buffer underflow and overflow. On the basis of simulation results it is concluded that Scheme IVb performs better than the others.
with buffer sizes of 1587 bits, and 1524 bits for the "Girl" and "Testcard" pictures respectively. A 4K bit RAM is therefore recommended as a suitable buffer for such adaptive transform image coding decoding system. A further control strategy employing coarse quantization of normalised transform coefficients during periods when the buffer is close to overflow has also been investigated (Section 5.3.4.5). From the comparison of reconstructed images obtained for "classification map control" and coarse quantization techniques, it is found that the results are subjectively quite similar (see Figures 5.44 (a,b) and 5.48 - 49). Both systems are considered to be useful and simple additions to adaptive transform coding image compression schemes.

8.6 ADAPTIVE TRANSFORM CODING OF MOVING IMAGES WITH VARIABLE BIT RATE BUFFERING

In adaptive transform image coding of still pictures the desired average bit rate is usually achieved by iteratively adjusting the distortion parameter. In the coding of moving images alternative methods can be employed. Five schemes were investigated to control the bit rate over the channel. In Scheme I the same value of distortion parameter is used for all classes and frames. A similar fixed distortion parameter approach is used in Scheme II, but here each class is assigned a different constant value. In Scheme III the value of distortion parameter was determined for the most active class, considering the maximum total
number of bits within this class as an approximate indicator of a given bit rate. The distribution of the total number of bits in each class is then corrected by setting the distortion parameter adaptively according to the instantaneous total number of bits within the second class (see Equation 6.9). This operation is repeated for the third and fourth classes. Scheme IV is a combination of the first and third schemes, where the distortion parameter is fixed for all classes and all frames as in Scheme I, and the instantaneous total numbers of bits for successive classes are compared. If the instantaneous total number in the present class is greater than the weighted instantaneous total for the previous class, then one bit is removed from the present class location (Equation 6.10). In Scheme V the interframe activity is considered where the activity of each block in the present frame is compared with that of the corresponding block of the previous frame. The distortion parameter is computed for the present block according to this comparison and the initialized value, and then the bit assignment carried out according to Scheme IV. All five schemes were used to code an eight frame moving sequence with Scheme V having the best overall performance - an average rate equal to 1 bit per element and 0.08% NMSE. The classification map control strategy for preventing buffer underflow and overflow is incorporated within Schemes III, IV, and V with maximum buffer size of about 3500 bits including overhead transmission through the buffer. In this case an 8 K bit RAM is suggested for use as a buffer store.
In practice the application of adaptive transform coding to real transmission channels involves an economic tradeoff in system design, balancing picture quality, circuit complexity, bit rate, and error performance. For the further reduction of system complexity and bit rate two techniques have been investigated, where the number of calculations and bit rate are reduced in the first and the system complexity and buffer size reduced in the second.

In the first scheme, the classification maps for successive frames are compared to estimate inter-frame energy. If they differ by less than a predefined threshold then the current classification map and bit allocation matrices are used for the succeeding frame, thus reducing the number of computational operations. In practice by choosing a suitable threshold one third of the total number of calculations may be eliminated. If in this case the present frame is not transmitted but the previous one is repeated in the receiver it is possible to achieve a transmission rate of 0.34 bits per element.

In the second scheme, intra-frame energy estimation is carried out at the receiver. The estimated variance of each transform coefficient along a zig-zag scanning path is found. The number of bits for each coefficient is then computed (proportional to the logarithm of the estimated variance within the scanned block). The computed number of
bits is then used for data extraction from the buffer, for image reconstruction and inverse normalization. The signal is then inverse transformed and displayed. This technique does not need any overhead information, but results in a reduction in buffer size. The simulated buffer size in this scheme is equal to about 200 bits at a rate of 1.2 bits per element. However, this scheme is not fully adaptive, which results in a relatively higher NMSE than the fully adaptive schemes which have been examined earlier in this work (see Table 7.II). Using such a scheme the overhead information requirement is eliminated and the system complexity reduced at the expense of higher transmission rate, as is to be expected.

8.8 SUGGESTIONS FOR FURTHER WORK

It is suggested that any further investigations in relation to adaptive transform coding systems using activity indices should be based on the new classifier which has been developed to reduce system complexity. Furthermore, classification map control to prevent buffer underflow and overflow is believed to be practically applicable and attractive because of its simplicity. In Scheme I of Chapter VII, the classification procedures have been omitted for the present frame whenever the difference between classification maps is lower than the predetermined threshold. For further investigations, the transformation operation within the coder for the present frame can be omitted too, to achieve a
substantial computational reduction, which will also result in a reduction in the system delay for low bit rate transmission. In addition, for the further investigation of buffering techniques using classification map control, an additional channel could be provided for the overhead information instead of using the variable input fixed output rate buffer. In this way a substantial reduction in buffer size can be achieved.
APPENDIX I

Some properties of matrices.

(a) A square matrix possesses an inverse if its determinant is non-zero. Such matrices are called non-singular.

(b) A real square matrix \([T]\) is called symmetric if \([T]^T = [T]\) where \([T]^T\) is the transpose of \([T]\).

\([T]\) is called orthogonal if \([T]^T[T] = [I]\) where \([I]\) is the identity matrix. Therefore a real square matrix is both symmetric and orthogonal if \([T]^{-1} = [T]\).

(c) A complex square matrix \([C]\) is called Hermitian if \([C]^* = [C]\) where \([C]^*\) is the complex conjugate of \([C]\).

\([C]\) is unitary if \([C]^*T[C] = I\).

Therefore a complex square matrix that is both Hermitian and unitary, \([C]^{-1} = [C]\).
APPENDIX II

C SIMULATOR FOR AN ADAPTIVE TRANSFORM IMAGE CODING - DECODING
C SYSTEM USING ACTIVITY INDEX WITH VARIABLE INPUT - FIXED
C OUTPUT TRANSMITTER BUFFER AND FIXED INPUT VARIABLE OUTPUT
C RECEIVER BUFFER WITH THE CONTROL OF CLASSIFICATION MAP
C INPUTS: L = ORIGINAL DATA
C G = TRANSFORM COEFFICIENTS
C MP = CLASSIFICATION MAP
C ICLASS = NUMBER OF CLASS
C QLAP1 = QUANTIZATION LEVELS INPUT
C QLAP2 = QUANTIZATION LEVELS OUTPUT
C IBIT = BIT ALLOCATION MATRICES
DIMENSION X(16,16),G(256),L(256),XX(16,256),Y(16,16),W(500),
1 YY(16,16),B(16,16),IC(256),SI(256),TH(256),GG(16)
DIMENSION MP(16,16),ICLASS(4),MAP(16,16),IBIT(16,16,4)
DIMENSION CONTENT(256)
DIMENSION IBIT(8)
DIMENSION CONTENTR(256)
DIMENSION IBUFF(4000)
DIMENSION IBUFF2(5000)
DIMENSION A(16,16)
DIMENSION IBY(8)
DIMENSION ICLIP(500)
DIMENSION COSIN(16,16)
DIMENSION MCLIP(8,256)
COMMON QLAP1(127),QLAP2(127)
DIMENSION IBIT(16,16,4)
C=7
DCMAX=4000
ILRX=1
BR=1
JJ=0
N=16
N1=256/N
PI=4.0*ATAN(1.0)
READ(5,343)(QLAP1(I),I=1,127)
READ(5,343)(QLAP2(I),I=1,127)
343 FORMAT(10F7.5)
DO 344 I=1,127
WRITE(2,345) QLAP1(I), QLAP2(I)
344 CONTINUE
345 FORMAT(1H ,2(F7.5,1DX))
DO 331 I=1,16
DO 340 K=1,16
COSIN(K,I)=COS(PI*(K-1)*(I-0.5)/N)
340 CONTINUE
331 CONTINUE
DO 15 K=1,4
READ(5,206)ICLASS(K)
READ(5,207)((MP(I,J),J=1,16),I=1,16)
207 FORMAT(72I1)
206 FORMAT(I1)
DO 16 IK=1,16
DO 16 IJ=1,16
MAP(IK,IJ)=MAP(IK,IJ)+MP(IK,IJ)
16 CONTINUE
READ(5,202)((IBIT(J,I,K),I=1,16),J=1,16)
202 FORMAT(16I1)
CONTINUE  
DO 901 K=1,4  
IBIT(1,1,K)=IBIT(1,1,K)  
DO 901 I=1,N  
DO 901 J=1,N  
IF(I+J.EQ.2)GO TO 901  
IF(IBIT(I,J,K).LT.4)GO TO 901  
IBIT(I,J,K)=IBIT(I,J,K)-1  
901 CONTINUE  
DO 3 K=1,16  
WRITE(2,203)(MAP(K,I),I=1,16)  
3 CONTINUE  
203 FORMAT(1HD,16(I1,3X))  
DO 4 K=1,4  
WRITE(2,204)ICLASS(K)  
DO 211 I=1,16  
WRITE(2,205)(IBIT(I,J,K),J=1,16)  
DO 211 NL=1,16  
211 IIBIT=IIBIT+IBIT(I,NL,K)  
WRITE(2,212)IIBIT  
IIBIT=0  
4 CONTINUE  
204 FORMAT(1HD,'CLASS',I1)  
205 FORMAT(1H,16(I1,4X))  
212 FORMAT('NO. OF BITS FOR A BLOCK=',I10)  
DO 110 IL=1,100  
IF(IL.GT.16)GO TO 312  
312 DO 120 LI=1,16  
IF(INAX.GT.2000)L000P=1  
IF(IL.GT.16) GO TO 315  
NWK=16*(IL-1)+LI
DO 1 I=1,N
READ (1)G
DO 2 K=1,16
VV(I,K)=G(K)
2 CONTINUE
1 CONTINUE

I=COUNT2+ICOUNT
COUNT=0.0

C ZX IS A FLAG FOR COARSE QUANTIZATION SCHEME
ZX=0
IF(IDIFF.GT.5000)ZX=1
MAPP=MAP(IL,LI)

C THE CONTROLLER
IF(MUP.EQ.MAPP.AND.IDIFF.GT.1500.AND.MAPP.LE.2)MAPP=MAPP+1
IF(IDIFF.LT.256.AND.MAPP.GT.1)MAPP=MAPP-1
IF(IDIFF.GT.1400.AND.MAPP.LT.4)MAPP=MAPP+1
MUP=MAP(IL,LI)

C THE CLASS ACTIVITY CODE WORD IS FED TO THE BUFFER
IF(INPADD.EQ.4DDD)INPADD=0
INPADD=INPADD+1
IBUFF(INPADD)=1
IF(MAPP.LT.3)IBUFF(INPADD)=0
IF(INPADD.EQ.4000)INPADD=0
INPADD=INPADD+1
IBUFF(INPADD)=1
IF(MAPP.EQ.1)IBUFF(INPADD)=0
IF(MAPP.EQ.3)IBUFF(INPADD)=0
IF(INPADD.EQ.4000)INPADD=0
INPADD=INPADD+1
IBUFF(INPADD)=0
IF(ZX.EQ.1)IBUFF(INPADD=1
DO 8 IK=1,16
DO 19 IJ=1,16

8 Continue
IF(IK+IJ).EQ.2) GO TO 10
IB=IBIT(IK,IJ,MAPP)
IF(ZX.EQ.1)IB=IBIT(IK,IJ,MAPP)
IF(IB.EQ.0)YY(IK,IJ)=0.0
IF(ABS(B(IK,IJ).GT.1.0)ICOUNT1=ICOUNT1+1
ICCOU=ICOUNT1
CON=C*(2.0**((IB-1))
C NORMALIZATION OPERATION
B(IK,IJ)=YY(IK,IJ)/CON
IF(IB.EQ.0)GO TO 444
MCLIP(IB,KUK)=MCLIP(IB,KUK)+1
C QUANTIZATION OPERATION
CALL QUALAP(B(IK,IJ),IB,IQ)
CALL ATBDC(IQ,IBI)
IF(B(IK,IJ).LT.0.0)IBI(9-IBI)=1
WRITE(2,888)IBI,IB
GO TO 445
10 CC=DMAX/255
IB=IBIT(IK,IJ,MAPP)
IF(ZX.EQ.1)IB=IBIT(IK,IJ,MAPP)
IQ=NINT(YY(1,1)/CC)
CALL ATBDC(IQ,IBI)
B(IK,IJ)=IQ*CC
WRITE(2,888)IBI,IB
888 FORMAT(1H,10I2,2X,I10)
55 FORMAT(1H,12D1)
C BINARY DATA FED TO THE TRANSMITTER BUFFER
445 DO 500 IE=1,IB
IF(INPADD.EQ.4DDD)INPADD=D INPADD=INPADD+1
INPADD=INPADD+1
IBUFF(INPADD)=IBI(5-IE)
500 CONTINUE
444 IF(OUTADD.GE.4000) OUTADD=OUTADD-4000.0
C DELAY IN TRANSMITTER BUFFER
   IF(KUK.LT.4) GO TO 19
C TRANSMITTER BUFFER OUTPUT
   OUTADD=OUTADD+BR
   IDIFF=INPADD-OUTADD
   IF(IDDIFF.LT.0) IDDIFF=IDDIFF+4000
   IF(IDDIFF.GT.0) GO TO 509
C INSERTER
   INSERT=INSERT+150
   WRITE(2,9999) INSERT
9999 FORMAT(1H,'NO OF BITS INSERTED=',I10)
   DO 502 KUF=1,150
   IF(INPADD.EQ.4000) INPADD=0
   INPADD=INPADD+1
   IDIFF=IDIFF+1
   IBUFF(INPADD)=1
502 CONTINUE
509 CONTINUE
315 IF(IL-17)332,333,334
333 IF(LI.NE.1) GO TO 334
337 IDIFF=IDIFF-1
   IF(OUTADD.GE.4000.0) OUTADD=OUTADD-4000.0
   OUTADD=OUTADD+BR
   ILAFL=1
   IDIFF=INPADD-OUTADD
332 IOUTADD=OUTADD
C RECEIVER BUFFER INPUT
   RXIN=RXIN+BR
   IF(RXIN.GT.5999.0) RXIN=RXIN-5999.0
   INRX=RXIN
   IF(INRX.EQ.ISTOR) GO TO 19
ISTOR=INRX
IBUFF2(INRX)=IBUFF(IDOUTADD)
IF(IBUFF(IDOUTADD)=504,504,505
C INSERTER EXCLUDER AT THE RECEIVER
I=ICRX=ICRX+1
IF(I=ICRX=.NE.150)GO TO 19
RXIN=RXIN-150.0
I=ICRX=0
IF(I=ILAFL.EQ.0)GO TO 336
IF(I=IDIFF.GT.0)GO TO 337
336 CONTINUE
19 CONTINUE
8 CONTINUE
IF(I=KUK.GT.256)GO TO 334
COUNTR(KUK)=KUK
CONTENT(KUK)=IDIFF
ICLIP(KUK)=ICOUNT
WRITE(2,309)((B(IK,IJ),IK=1,16),IJ=1,16)
WRITE(2,307)(IBUFF(KLM),KLM=1,INPADD
WRITE(2,308)(IBUFF2(KLM),KLM=1,INRX)
LOOP=LOOP+1
IF(LOOP.LT.3)GO TO 311
308 FORMAT(1H,6DI2)
C DELAY AT RECEIVER BUFFER
334 IF(LOOP.EQ.0)GO TO 311
LIRX=LIRX+1
IF(LIRX.EQ.17)GO TO 321
IF(I=ILRX+LIRX.EQ.2)GO TO 320
DO 318 I=1,N
READ(3)L
DO 319 K=1,256
XX(I,K)=L(K)
319  CONTINUE
318  CONTINUE
320  DO 121 I=1,N
   DO 121 K=1,N
      K1=(LIRX-1)*N+K
      X(I,K)=XX(I,K1)
   121  CONTINUE
321  IF(LIRX.LT.17) GO TO 508
   LIRX=1
   LIRX=LIRX+1
   IF(LIRX.GT.16) GO TO 314
   DO 324 I=1,N
   READ(3) L
   DO 325 K=1,256
      XX(I,K)=L(K)
   325  CONTINUE
   DO 326 I=1,N
   DO 327 K=1,N
      K1=(LIRX-1)*N+K
      X(I,K)=XX(I,K1)
   327  CONTINUE
326  CONTINUE
508  IF(IOUTRX.EQ.5999) IOUTRX=0
C  RECEIVER BUFFER OUTPUT
   IOUTRX=IOUTRX+1
   MAP1=IBUFF2(IOUTRX)
   IF(IOUTRX.EQ.5999) IOUTRX=0
   IOUTRX=IOUTRX+1
   MAP2=IBUFF2(IOUTRX)
   MAPPP=(MAP2+(2*MAP1)+1)
   IF(IOUTRX.EQ.5999) IOUTRX=0
   IOUTRX=IOUTRX+1
ZZX=1BUFF2(IOUTRX)
DO 313 KRX=1,16
DO 313 MRX=1,16
313 A(KRX,MRX)=0.0
DO 304 KRX=1,16
DO 305 MRX=1,16
IY=IBIT(KRX,MRX,MAPPP)
IF(ZZX.EQ.1)IY=IBIT(KRX,MRX,MAPPP)
IA=0
IF(IY.EQ.0)GO TO 305
DO 302 IZ=1,8
302 IBY(IZ)=0
DO 303 IZ=1,IY
IF(IOUTRX.EQ.5999)IOUTRX=0
IOUTRX=IOUTRX+1
IBY(9-IZ)=1BUFF2(IOUTRX)
303 CONTINUE
IF(KRX+MRX.EQ.2)GO TO 348
SIGN=1.0
IF(IBY(9-IY).EQ.0)SIGN=1.0
IBY(9-IY)=0
C DECODING OPERATION
348 IA=IBY(8)+(2*IBY(7))+(4*IBY(6))+(8*IBY(5))+(16*IBY(4))+(32*IBY(3))
1)+(64*IBY(2))+(128*IBY(1))
WRITE(2,317)IBY,IY
317 FORMAT(1H ,4DX,1DI2,2X,I10)
C RECONSTRUCTION
IF(KRX+MRX.EQ.2)GO TO 306
N11=2**((IY-1)
N12=N11-1
IF(IA.GE.N12)IA=N12
IF(N11.EQ.1)IA=0
CON=C*(2.0**(IV-1))
A(KRX,MRX)=QLAP2(N11+IA)*CON*SIGN
GO TO 305

306 CC=DMAX/256
A(KRX,MRX)=IA*CC
305 CONTINUE
304 CONTINUE
WRITE(2,309)((A(KRX,MRX),MRX=1,16),KRX=1,16)
309 FORMAT(1H ,16F7.2)
IF(INPADD.GE.0)GO TO 17
WRITE(2,77)INPADD
77 FORMAT(1H ,'BUFFER ADDRESS IS NEGATIVE',I10)
17 CONTINUE
IX=CONTENT(KUK)
WRITE(2,213)INPADD,IOUTADD
213 FORMAT('INPADDRESS=' ,I10,5X,'OUTPUTADDRESS=' ,I10)
WRITE(2,749)IL,LI,IX,MAP(IL,LI),MAPP
749 FORMAT(1H120X,IL=' ,I3,5X,LI=' ,I3,5X,'CONTENT=' ,I8,5X,'ICLASS=' ,I1
CONTENT2=CONTENT2+CONTENT(KUK)
IF(ILRX+LIRX.LT.2)GO TO 120
ICONTRX=INRX-IDUTRX
WRITE(2,214)ICONTRX,INRX,IOUTRX
214 FORMAT(1H 'CONTENT OF REC BUFF=' ,I6,5X,'INRX=' ,I10,'IOUTRX=' ,I10)
WRITE(2,316)ILRX,LIRX,MAP(ILRX,LIRX),MAPP
316 FORMAT(1H120X,'ILRX=' ,I3,5X,'LIRX=' ,I3,5X,'ICLASS=' ,I3,5X,'ICLASS=' ,I3)
C INVERSE TRANSFORMATION
DO 31 K=1,N
DO 30 II=1,N
Y(K,II)=0.0
DO 32 KK=1,N
CN=1.0
IF(KK.EQ.1)CN=1.0/SQRT(2.0)
Y(K,II)=Y(K,II)+A(K,KK)*CN*COS(KK,II)
32 CONTINUE
30 CONTINUE
31 CONTINUE
DO 33 I=1,N
DO 34 K=1,N
A(I,K)=0.0
DO 35 II=1,N
CM=1.0
IF(II.EQ.1)CM=1.0/SQRT(2.0)
A(I,K)=A(I,K)+CM*Y(II,K)*COS(II,I)
35 CONTINUE
A(I,K)=A(I,K)*2/N
IF(A(I,K).GT.255.0)A(I,K)=255.0
IF(A(I,K).LT.0.0)A(I,K)=0.0
GG(K)=A(I,K)
34 CONTINUE
C WRITE(2,9)(A(I,K),K=1,N
9 FORMAT(1HO,16(F6.2,1X))
33 CONTINUE
C NORMALIZED MEAN SQUARE ERROR OF EACH BLOCK
SIGMA=0.0
SIGMA1=0.0
DO 75 I=1,N
DO 75 J=1,N
K1=(LIRX-1)*N+J  
XX(I,K1)=A(I,J)  
A(I,J)=A(I,J)-X(I,J)  
SIGMA=SIGMA+A(I,J)*A(I,J)  
SIGMA1=SIGMA1+X(I,J)**2  
SIGMA1=SIGMA1+(X(I,J)-B6)**2  
CONTINUE  
SIGMA=SIGMA/SIGMA1  
SI(KUK)=SIGMA  
IREC=16*(ILRX-1)+LIRX  
IC(IREC)=ICLIP(IREC)  
SI(IREC)=SIGMA  
CONTINUE  
120 CONTINUE  
IF(LOOOP.EQ.0)GO TO 110  
DO 200 I=1,N  
DO 201 K=1,256  
L(K)=NINT(XX(I,K))  
201 CONTINUE  
WRITE(4,210)(L(K),K=1,256)  
210 FORMAT(256I10)  
200 CONTINUE  
110 CONTINUE  
CONTINUE  
CONTENT3=CONTENT2/256  
WRITE(2,12)CONTENT3  
12 FORMAT('CONTENT3=',F10.2)  
SIGMA=0.0  
DO 112 I=1,256  
SIGMA=SIGMA+SI(I)  
WRITE(2,113)I,TH(I),SI(I),IC(I)  
WRITE(2,113)I,TH(I),SI(I),IC(I),(MCLIP(K,I),K=1,8)  
112 CONTINUE
C AVERAGE NORMALIZED MEAN SQUARE ERROR FOR ALL PICTURE
SIGMA=SIGMA/256
WRITE(2,5)SIGMA

5 FORMAT(1HO,'AV NOR M S E=',F9.7)
113 FORMAT(1HO,I3,5X,2(F10.5,5X),I5,10X,8(I3,1X))
WRITE(2,330)ICOUNT2,ICOUNT1

330 FORMAT(1H,'ICOUNT2=',I10,10X,'ICOUNT1=',I10)

CALL C1051N
CALL GRAF(COUNTR,CONTENT,256,0)
CALL DEVEND
STOP
END
SUBROUTINE QULAP(B,IB,IQ)

    THIS SUBROUTINE QUANTISES THE NORMALIZED TRANSFORM
    COEFFICIENTS WITH A LAPLACIAN PDF

    THE INPUTS: B=DATA TO BE QUANTIZED
    IB=ORDER OF QUANTIZER

    THE OUTPUTS:IQ=NUMBER OF QUANTIZATION LEVEL

    B=QUANTIZED DATA

COMMON QLAP1(127),QLAP2(127)
M=2**IB
IQ=0
N2=M-1
IF(IB.EQ.1)GO TO 11
V=ABS(B)
N1=(M/2)+1
DO 100 I=N1,N2
  IF(V.GE.QLAP1(I-1).AND.V.LT.QLAP1(I))GO TO 33
  IQ=IQ+1
  IF(V.GE.QLAP1(N2))V=QLAP2(N2)
100 CONTINUE
GO TO 22
33 V=QLAP2(I-1)
GO TO 22
11 V=QLAP2(I)
22 IF(B.GT.0.0)GO TO 23
   Y=-V
23 B=Y
RETURN
END
SUBROUTINE ATBDC(IQ,IBI)
C THIS SUBROUTINE IS CODING THE NUMBER OF QUANTIZATION
C LEVELS TO A BINARY CODE WORD
DIMENSION IBI(8)
DO 2 I=1,8
2 IBI(I)=0
DO 1 I=1,8
J=9-I
RD=(IQ/2.0)-(IQ/2)
IQ=IQ/2
IF(RD.NE.0.0)IBI(J)=1
1 CONTINUE
IF(IQ.EQ.0)RETURN
WRITE(2,222)
222 FORMAT('NO. REQUIRE MORE THAN 8 BITS')
RETURN
END
FINNISH
SUBROUTINE TRAN2(X,B,N)
C COMPUTES THE TWO-DIM. TRANSFORM
C X = NXN INPUT ARRAY (ORIGINAL DATA)
C B = NXN TRANSFORM KERNEL
DIMENSION X(16,16), B(16,16)
DIMENSION Z(256)
F=FLOAT(N)
C EACH COLUMN OF X IS COPIED TO A DUMMY VECTOR Z, 
C TRANSFORMED, AND STORED IN THE SAME COLUMN OF X.
DO 10 J=1,N
DO 5 L=1,N
5 Z(L)=X(L,J)
DO 10 K=1,N
X(K,J)=0.0
DO 10 L=1,N
X(K,J)=X(K,J)+B(K,L)*Z(L)
10 CONTINUE
C EACH ROW OF X IS COPIED TO A DUMMY VECTOR Z, 
C TRANSFORMED, AND STORED IN THE SAME ROW OF X
DO 30 I=1,N
DO 20 L=1,N
20 Z(L)=X(I,L)
DO 30 K=1,N
X(I,K)=0.0
DO 30 L=1,N
X(I,K)=X(I,K)+B(K,L)*Z(L)
30 CONTINUE
WRITE(2,15)((X(I,K),K=1,16),I=1,16)
15 FORMAT(1HO,16(F6.2))
RETURN
END
SUBROUTINE BCOS(B,N)
C     COMPUTES THE COSINE BASIS FUNCTIONS
C     N=ORDER OF TRANSFORM
C     B=COSINE BASIC FUNCTION MATRIX
DIMENSION B(N,N)
PI=3.1415926
F=FLOAT(N)
DO 190 I=1,N
   IA=I-1
   AN=FLOAT(IA)
   CD=SORT(2.0/N)
   IF(IA.EQ.0)CD=SORT(1.0/N)
   DO 191 J=1,N
      IS=J-1
      BB=FLOAT(IS)
      191 B(I, J)=CD*COS((BB+0.5)*AN*PI/F)
190 CONTINUE
WRITE(2,4000)(( B(I, J),J=1,16), I=1,16)
4000 FORMAT(1H ,16F7.4)
RETURN
END
APPENDIX III

ADAPTIVE TRANSFORM CODING - DECODING SYSTEM SIMULATOR WITH TRANSMITTER AND RECEIVER BUFFERS FOR MOVING IMAGE SEQUENCE.

DEFINE FILE3(256,256,IN)
DEFINE FILE1(256,256,IN)
DIMENSION L(256)
DIMENSION IX(64,64),G(256),XX(16,256),Y(64,64),W(500)

YY(16,16),X(64,64),IC(256),SI(256),TH(256),GG(16)
DIMENSION MP(16,16),ICLASS(4),MAP(16,16),IBIT(16,16,4)
DIMENSION IBIT(8)
DIMENSION CONTENT(192),COUNTR(192)
DIMENSION IBUFF(4000),IBUFF2(6000)
DIMENSION MAPX(4,6,8)
DIMENSION IBITX(16,16,4)
DIMENSION A(16,16)
DIMENSION IBY(8)
DIMENSION ICLIP(500)
DIMENSION L1(256)
DIMENSION COSIN(16,16)
DIMENSION MAPD(16,16)
DIMENSION MAPCHNG(16)
DIMENSION X(16,16)

ILRX=4
LIRX=6
BR=1.6
\begin{verbatim}
JJ=0
N=16
N1=256/N
MAPCONT=0
PI=4.0*ATAN(1.0)
DOMAX=3000
C=7
DO 331 I=1,16
DO 340 K=1,16
  COS(K,I)=COS(PI*(K-1)*(I-0.5)/N)
340  CONTINUE
331  CONTINUE
DO 344 IFRAME=1,8
DO 775 K=1,4
DO 775 I=1,6
  MAPD(K,I)=MAP(K,I)
  IF(MAPCONT.EQ.1)GOTO 776
  MAPCHNG(IFRAME)=1
  CALL COSI(VY,ICLASS,MAP,IBIT,IFRAME)
776  MAPCONT1=MAPCONT
  MAPCONT=0
  MAPCON2=1
  IF(MAPCONT.NE.MAPCONT1)MAPCON2=0
 ICI=0
  DO 777 K=1,4
  DO 777 I=1,6
  IF(MAP(K,I).NE.MAPOD(K,I))ICI=ICI+1
  IF(ICI.LT.8)MAPCONT=1+MAPCON2
  DO 1001 J=1,4
  DO 1001 I=1,16
  DO 1001 K=1,16
  IQ=IBIT(I,K,J)
  IF((I+K).EQ.2)IQ=IQ-1
\end{verbatim}
CALL ATEDC(IQ,IBI,IB)
DO 1001 IE=1,3
IF(INPADD.EQ.4000)INPADD=0
INPADD=INPADD+1
IBUFF(INPADD=IBI(9-IE)
1001 CONTINUE
DO 3 K=1,16
WRITE(2,203)(MAP(K,I),I=1,16)
WRITE(2,8888)IBUFF
8888 FORMAT(1H0,6012)
3 CONTINUE
DO 720 LMK=1,4
DO 720 MKL=1,6
MAPX(LMK,MKL,IFRAME)=MAP(LMK,MKL)
720 CONTINUE
203 FORMAT(1H0,16(I1,3X))
DO 4 K=1,4
WRITE(2,204)ICLASS(K)
DO 211 I=1,16
WRITE(2,205)(I BIT(I,J,K),J=1,16)
DO 211 NL=1,16
211 IIBIT=IIBIT+IBIT(I,NL,K)
WRITE(2,212)IIBIT
IIBIT=0
4 CONTINUE
204 FORMAT(1H0,'CLASS',I1)
205 FORMAT(1H0,16(I1,4X))
212 FORMAT('NO. OF BITS FOR A BLOCK=',I10)
NNN=4
IF(IFRAME.EQ.8)NNN=1000
DO 110 IL=1,NNN

DO 120 LT=1,6
   IF(INRX.GT.4000)LOOP=1
   IF(IL.GT.4)GO TO 315
   KUK=6*(IL-1)+LT
   IFRA=(IFRAME-1)*24
   IBLOCK=IFRA+KUK
   WRITE(2,718)IBLOCK
5718 FORMAT(1H , 'NUMBER OF BLOCK=',I4)
   NUMBER=0
   READ(1'IBLOCK)G
   DO 1 I=1,N
      DO 2 K=1,16
         NUMBER=NUMBER+1
      VV(I,K)=G(NUMBER)
   2 CONTINUE
1 CONTINUE
   I COUNT2=ICOUNT2+ICOUNT
   ICOUNT=0.0
   MAPP=MAP(IL,LI)
   IF(MUP.EQ.MAPP.AND.IDIFF.GT.3500.AND.MAPP.LE.2)MAPP=MAPP+1
   IF(IDIFF.LT.256.AND.MAPP.GT.1)MAPP=MAPP-1
   IF(IDIFF.GT.3000.AND.MAPP.LT.4)MAPP=MAPP+1
   MUP=MAP(IL,LI)
   IF(INPADD.EQ.4000)INPADD=0
   INPADD=INPADD+1
   IBUFF(INPADD)=1
   IF(MAPP.LT.3)IBUFF(INPADD)=0
   IF(INPADD.EQ.4000)INPADD=0
   INPADD=INPADD+1
   IBUFF(INPADD)=1
   IF(MAPP.EQ.1)IBUFF(INPADD)=0
IF(MAPP.EQ.3)BUFF(INPADD)=0
DO 8 IK=1,16
DO 19 IJ=1,16
IF(ABS(2)) GO TO 10
IB=1BIT(IK,IJ,MAPP)
IF(IB.EQ.0)YY(IK,IJ)=0.0
CON=C*(2.0**(IB-1))
B(IK,IJ)=YY(IK,IJ)/CON
IF(IB.EQ.0)GO TO 444
IF(B(IK,IJ).LT.0.0)B(IK,IJ)=0.0
CALL QUALAP(B(IK,IJ),IB,IQ)
CALL ATBDC(IQ,IB1,IB)
IF(B(IK,IJ).LT.0.0)IB(9-IB)=1
WRITE (2,888) IB1,IB
GO TO 445
10 CC=DCMAX/255
IB=1BIT(IK,IJ,MAPP)
IQ=NINT(YY(1,1)/CC)
CALL ATBDC(IQ,IB1,IB)
WRITE(2,888)IB1,IB
888 FORMAT(1H120I1)
55 FORMAT(1H120I1)
445 DO 500 IE=1,IB
IF(INPADD.EQ.4000)INPADD=0
INPADD=INPADD+1
IBUFF(INPADD)=IB(9-IE)
500 CONTINUE
BR1=BR+BR1
IBR1=IBR
IBR=BR1
IBR2=IBR-IBR1
444 DO 19 IE=1,IBR2
IF(DUTADD.GE.6000)DUTADD=DUTADD-6000.0
IF(IFRAME.EQ.1.AND.IBLOCK.LT.3)GO TO 19
DOUTADD=DOUTADD+1.0
IDIFF=INPADD-DOUTADD
IF(IDIFF.LT.0)IDIFF=IDIFF+4000
IF(IDIFF.GT.0)GO TO 509
INSERT=INSERT+150
WRITE(2,9999)INSERT
9999 FORMAT(1H,,'NNO OF ONES INSERTED=',I10)
DO 502 KUF=1,150
INPADD=INPADD+1
IDIFF=IDIFF+1
IBUFF(INPADD)=1
502 CONTINUE
509 CONTINUE
315 IF(IL-5)332,333,334
333 IF(IL.NE.1)GO TO 334
337 IDIFF=IDIFF-1
ILAFL=1
332 IOUTADD=DOUTADD
RXIN=RXIN+1.0
IF(RXIN.GT.5999.0)RXIN=RXIN-5999.0
INRX=RXIN
IF(INRX.EQ.ISTOR)GO TO 19
ISTOR=INRX
IBUFF2(INRX)=IBUFF(IOUTADD)
IF(IBUFF(IOUTADD))504,504,505
505 ICRX=ICRX+1
IF(ICRX.NE.150.AND.ILAFL.NE.1)GO TO 19
IF(ICRX.NE.150.AND.ILAFL.EQ.1)GO TO 555
RXIN=RXIN-150.0
504 ICRX=0
   IF(ILAFL.EQ.0)GO TO 336
555 IF(IDIFF.GT.0)GO TO 337
336 CONTINUE
19 CONTINUE
8 CONTINUE
   ICLIP(KUK)=ICOUNT
   IF(IABS.EQ.192)GO TO 334
   IABS=IABS+1
   COUNTR(IABS)=IABS
   CONTENT(IABS)=IDIFF
   WRITE(2,309)((B(IK,IJ),IK=1,16),IJ=1,16)
   WRITE(2,307)(IBUFF(KLM).KLM=1,INPAD)
   WRITE(2,308)(IBUFF2(KLM).KLM=1,INRX)
308 FORMAT(1H,6DI2)
334 IF(LOOP.EQ.0)GO TO 311
   IFINISH=IFINISH+1
   IF((LIRX+ILRX).LT.10)GO TO 319
   IFAM=IFAM+1
   DO 1003 J=1,4
   DO 1003 I=1,16
   DO 1003 K=1,16
   IF(IOUTRX.EQ.5999)IOUTRX=0
   IOUTRX=IOUTRX+1
   IQ1=IBUFF2(IOUTRX)
   IF(IOUTRX.EQ.5999)IOUTRX=0
   IOUTRX=IOUTRX+1
   IQ2=IBUFF2(IOUTRX)
   IF(IOUTRX.EQ.5999)IOUTRX=0
   IOUTRX=IOUTRX+1
   IQ3=IBUFF2(IOUTRX)
IQ=IQ1+2*IQ2+4*IQ3
IF((I+K).EQ.2)IQ=IQ+1
IBITX(I,K,J)=IQ
1003 CONTINUE
ILRX=1
LIRX=0
319 IF(LIRX.LT.6)GO TO 320
LIRX=0
ILRX=ILRX+1
320 LIRX=LIRX+1
ILRX=ILRX+1
NUMBER=0
READ(3'I BLRX)L
DO 318 I=1,16
DO 318 K=1,16
NUMBER=NUMBER+1
LX(I,K)=L(NUMBER)
318 CONTINUE
WRITE(2,350)((IX(I,K),K=1,16),I=1,16)
350 FORMAT(1H ,36I7)
IF(IOUTRX.EQ.5999)IOUTRX=0
508 IOUTRX=IOUTRX+1
MAP1=IBUFF2(IOUTRX)
IF(IOUTRX.EQ.5999)IOUTRX=0
IOUTRX=IOUTRX+1
MAP2=IBUFF2(IOUTRX)
MAPPP=(MAP2+(2*MAP1)+1)
DO 313 KRX=1,16
DO 313 MRX=1,16
313 A(KRX,MRX)=0.0
DO 304 KRX=1,16
DO 305 MRX=1,16
IV=IBIT(KRX,MRX,MAPP)
IA=0
IF(IV.EQ.0)GO TO 305
DO 302 IZ=1,8
302 IBY(IZ)=0
DO 303 IZ=1,IV
IF(IOUTRX.EQ.5999)IOUTRX=0
IOUTRX=IOUTRX+1
IBY(9-IZ)=IBUFF2(IOUTRX)
303 CONTINUE
IF(KRX+MRX.EQ.2)GO TO 348
SIGN=-1.0
IF(IBY(9-IV).EQ.0)SIGN=1.0
IBY(9-IV)=0
348 IA=IBY(8)+(2*IBY(7))+(4*IBY(6))+(8*IBY(5))+(16*IBY(4))+(32*IBY(3))
   +(64*IBY(2))+(128*IBY(1))
WRITE(2,317)IBY,IV
317 FORMAT(1H ,40X10I2,2X,I10)
IF(KRX+MRX.EQ.2)GO TO 306
N11=2**(IV-1)
N12=N11-1
IF(IA.GE.N12)IA=N12
IF(N11.EQ.1)IA=0
CON=C*(2.0***(IV-1))
A(KRX,MRX)=QLAP2(N11+IA)*CON*SIGN
GO TO 305
306 CC=DCMAX/256
A(KRX,MRX)=IA*CC
CONTINUE
CONTINUE
WRITE(2,309)((A(KRX,MRX),MRX=1,16),KRX=1,16)
FORMAT(1H,16F7.2)
IF(INPADD.GE.0)GO TO 17
WRITE(2,77)INPADD
77 FORMAT(1H,'BUFFER ADDRESS IS NEGATIVE',I10)
CONTINUE
ID=CONTENT(IABS)
WRITE(2,213)INPADD,IOUTADD
213 FORMAT(1H,'INADDRESS=',I10,5X,'OUTPUTADDRESS=',I10)
WRITE(2,749)IL,LI,ID,MAP(IL,LI),MAPP
739 FORMAT(1H0,' 20X',IIL='13,5X','IL='13,5X,'CONTENT='13,5X,'ICLASS='13,5X,'ICLASS='11)
CONTENT2=CONTENT2+CONTENT(KUK)
IF(ILRX+LI RX.LT.2)GO TO 120
I ConTrX=INRX-IDTrX
IF(I ConTrX.LT.0)I CONTrX=I CONTrX+5999
WRITE(2,214)I CONTrX,INRX,IOU TRX
214 FORMAT(1H,'CONTENT OF REC BUFF=',I6,5X,'INRX=',I10,'IOUTRX=',I10)
WRITE(2,316)ILRX,LIRX,MAPX(ILRX,LIRX,IFAM),MAPPP
316 FORMAT(1H0,20X,'ILRX=',I3,5X,'LIRX=',I3,5X,'ICLASS=',I3,5X,'
IICLASS=',I13)
DO 31 K=1,N
DO 30 II=1,N
V(K,II)=0.0
DO 32 KK=1,N
CN=1.0
IF(KK.EQ.1)CN=1.0/SORT(2.0)
V(K,II)=V(K,II)+A(K,KK)*CN*COSIN(KK,II)
30 CONTINUE
31 CONTINUE
   DO 33 I=1,N
   DO 34 K=1,N
      A(I,K)=0.0
   DO 35 II=1,N
      CM=1.0
      IF(II.EQ.1)CM=1.D/SQRT(2.0)
      A(I,K)=A(I,K)*CM*V(II,K)*COS(II,I)
   CONTINUE
   A(I,K)=A(I,K)*2/N
   IF(A(I,K).GT.255.0)A(I,K)=255.0
   IF(A(I,K).LT.0.0)A(I,K)=0.0
   GG(K)=A(I,K)
34 CONTINUE
   WRITE(2,9)(A(I,K),K=1,N)
9 FORMAT(1HD,16(F6.2,1X))
33 CONTINUE
   NUMBER=0
   DO 102 I=1,16
      DO 103 K=1,16
         NUMBER=NUMBER+1
         L1(NUMBER)=NINT(A(I,K))
      CONTINUE
103 CONTINUE
102 CONTINUE
   WRITE(4,I8.8)L1
   WRITE(5,1000)(L1(NUMBER),NUMBER=1,256)
1000 FORMAT(256I4)
   WRITE(2,104)L1
104 FORMAT(1H,16I6)
WRITE (2, 99)
99 FORMAT (1H0, /)
SIGMA = 0.0
SIGMA1 = 0.0
DO 75 I = 1, N
DO 75 J = 1, N
   K1 = (LIRX - 1) * N * J
   XV(I, K1) = A(I, J)
   A(I, J) = A(I, J) - IX(I, J)
   SIGMA = SIGMA + A(I, J) * A(I, J)
   SIGMA1 = SIGMA1 + IX(I, J)**2
75 CONTINUE
SIGMA = SIGMA / SIGMA1
SI(KUK) = SIGMA
IREC = 6 * (ILRX - 1) + LIRX
IQ(IREC) = ICLIP(IREC)
SI(IREC) = SIGMA
311 CONTINUE
120 CONTINUE
IF (LODDP.EQ.0) GO TO 110
DO 200 I = 1, N
   DO 201 K = 1, 256
      L1(K) = NINT(XX(I, K))
   201 CONTINUE
   WRITE (4, 210) (L1(K), K = 1, 256)
210 FORMAT (256I10)
200 CONTINUE
110 CONTINUE
314 CONTINUE
CONTENT3 = CONTENT2 / 24
WRITE (2, 12) CONTENT3
12 FORMAT(1H, 'CONTENT3=', F10.2)
    SIGMA = 0.0
    DO 112 I = 1, 24
        SIGMA = SIGMA + SI(I)
        WRITE(2, 113) I, TH(I), SI(I), IC(I)
    112 CONTINUE
    SIGMA = SIGMA / 24
    WRITE(2, 5) SIGMA
5 FORMAT(1HO, 'AV NOR M S F=', F9.7)
113 FORMAT(1HO, I3, 5X, 2(F10.5, 5X), I5)
    WRITE(2, 330) I COUNT2, I COUNT1
330 FORMAT(1H, 'ICOUNT2=', I10, 10X, 'ICOUNT1=', I10)
    WRITE(2, 719) COUNTR
    WRITE(2, 719) CONTENT
719 FORMAT(1H, 12F10.1)
344 CONTINUE
    CALL C1051N
    CALL GRAF(COUNTR, CONTENT, 192.0)
    CALL DEVEND
    DO 778 I = 1, 16
        WRITE(2, 779) I, MAPCHNG(I)
    778 CONTINUE
779 FORMAT(1H, I2, 5X, I2)
STOP
END
SUBROUTINE COSI(VV, ICL, MAP, IBIT, IFRAME)
C
THIS SUBROUTINE SIMULATES THE BIT ASSIGNMENT ALGORITHMS
DIMENSION Q(24)
DIMENSION IND(24)
DIMENSION MAP(16,16)
DIMENSION SIG(16,16)
DIMENSION VV(16,16)
DIMENSION SIGK(16,16)
DIMENSION Z(16,16,25), SIGMA(16,16)
DIMENSION X(16,16), Y(16,16)
INTEGER BIT(16,16,4)
DIMENSION ICL(4)
DIMENSION G(256)
DIMENSION DI(4)
DIMENSION DIS(16,16)
DIMENSION CIN(16,16), B(PRIME(16,16), BCLA(16,16)
N=16
C=5
DI(1)=5
DI(2)=4
DI(3)=4
DI(4)=6
D=5
FF=0.9
ALOG2=2*ALOG10(2.0)
CALL THR(VV, IFRAME, Q, IND, MAP)
103 FORMAT(1HO, I3, 3X)
DO 401 LOOP1=1,4
IFRA=(IFRAME=1)*24
ICLASS=LOOP1
D=DI(LOOP 1)
DO 1 MKM=1,6

.IBLOCK=IND(MKM+(LOOP1-1)*6)+1FRA
NUMBER=0
READ(1'I BLOCK)G
DO 4 I=1,16
DO 5 K=1,16
NUMBER=NUMBER+1
YY(I,K)=G(NUMBER)
5 CONTINUE
4 CONTINUE
WRITE(2,17)((YY(I,K),K=1,16),I=1,16)
MLK=6
DO 101 KLM=1,N
DO 102 LM=1,N
Z(KLM,LM,MKM)=YY(KLM,LM)
102 CONTINUE
101 CONTINUE
WRITE(2,33)II,VARI(I)
WRITE(2,17)G
1 CONTINUE
33 FORMAT('VARIANCE OF BLOCK',I5,'IS=',F10.5)
ICL(LOOP1)=ICLASS
DO 77 IM=1,MLK
DO 78 J=1,N
DO 79 K=1,N
YY(J,K)=Z(J,K,IM)
79 CONTINUE
78 CONTINUE
YY(1,1)=0.0
DO 18 I=1,N
DO 19 J=1,N
Y(I,J)=Y(I,J)+YY(I,J)
X(I,J)=X(I,J)+(YY(I,J)**2)
19 CONTINUE
18 CONTINUE
77 CONTINUE
   DO 20 I=1,N
   DO 21 J=1,N
   Y(I,J)=Y(I,J)/MLK
   X(I,J)=X(I,J)/MLK
21 CONTINUE
   WRITE(2,17)(Y(I,M),M=1,16)
20 CONTINUE
   BCL=0.0
   BCL1=0.0
   BCL2=0.0
   CI=0.0
   DO 23 K=1,N
   DO 23 I=1,K
   K1=K
   K2=K
   IF(I.NE.K)GO TO 409
   K1=1
   K2=K
409 DO 23 J=K1,K2
   SIGMA(I,J)=X(I,J)-(Y(I,J)**2)
   SIG(I,J)=SQRT(SIGMA(I,J))
   IF(SIGMA(I,J),EQ.0.0)SIGMA(I,J)=1.0E-20
   D=0.1
   CCI=I*I+J*J
   CI=SQRT(CCI)
   IF(CI.LT.8)GO TO 410
   IF(BCL.GT.BCL1*FF)D=D1*(1+CI/100)*BCL/(BCL1*FF)
   IF(LOOP1.NE.1)GO TO 410
   IF(I+J.LT.20)GO TO 410
   IF(BCL2.LT.400)D=0.9*D
410 BIT(I,J,LOOP1)=NINT(ALOG10(SIGMA(I,J)/D)/ALOG2)
IF(BCL.GT.BCL.1+F2)BIT(I,J,LOOP1)=BIT(I,J,LOOP1)-1
IF(BIT(I,J,LOOP1).LT.0)BIT(I,J,LOOP1)=0
IF(BIT(I,J,LOOP1).GT.7)BIT(I,J,LOOP1)=7
BCL2=BIT(I,J,LOOP1)+BCL2
IF(LOOP1.EQ.1)GO TO 22
BCL=BIT(I,J,LOOP1-1)+BCL
BCL=BIT(I,J,LOOP1)+BCL
22 CONTINUE
DIS(I,J)=0
CIN(I,J)=CI
BCL1A(I,J)=BCL1
BCLA(I,J)=BCL
23 CONTINUE
WRITE(2,17)((DIS(I,J),J=1,16),I=1,16)
WRITE(2,17)((CIN(I,J),J=1,16),I=1,16)
WRITE(2,17)((BCL1A(I,J),J=1,16),I=1,16)
WRITE(2,17)((BCLA(I,J),J=1,16),I=1,16)
WRITE(2,17)((SIGMA(I,J),J=1,16),I=1,16)
DO 44 I=1,N
DO 45 J=1,N
SIGK(I,J)=C*2**(BIT(I,J,ICLASS)-1)
45 CONTINUE
WRITE(2,17)(SIGK(I,J),J=1,16)
44 CONTINUE
17 FORMAT(1HD,16F7.2)
BIT(1,1,ICLASS)=8
DO 110 I=1,N
WRITE(2,111)(BIT(I,J,LOOP1),J=1,16)
DO 110 J=1,N
ICOUNT=ICOUNT+BIT(I,J,LOOP1)
110 CONTINUE
111 FORMAT(1HD,16I3)
WRITE(2,112) ICOUNT

112 FORMAT(1HO,I10)
ICOUNT=0
R1=0.0
R2=0.0
R31=0
R32=0
MLK=0
DO 402 J=1,16
DO 402 I=1,16
X(I,J)=0
Y(I,J)=0
DO 403 K=1,N
DO 403 LK=1,N
DIS(K,LK)=0.0
403 CONTINUE
402 CONTINUE
401 CONTINUE
RETURN
END
SUBROUTINE ATADC(IQ,I8)
DIMENSION IBI(11)
DIMENSION IBI(8)
DO 2 I=1,8
  IBI(I)=0
DO 1 I=1,8
  J=9-I
  RD=(IQ/2.0)-(IQ/2)
  IQ=IQ/2
  IF(RD.NE.0.0)IBI(J)=1
1 CONTINUE
RETURN
END

SUBROUTINE THR(VV,FRAME,Q,IND,MAP)
C THIS SUBROUTINE IS THE CLASSIFIER FOR MOVING IMAGE SEQUENCE.
C IT IS TO FIND THE ACTIVITY INDEX FOR EACH 24 TRANSFORMED
C BLOCKS(VARIANCE OF A.C COEFFICIENTS OR THEIR MAGNITUDES),
C THEN REORDER, AND DIVIDE FOR FOUR ACTIVITY CLASSES.
DIMENSION MAP(16,16)
DIMENSION VV(16,16),VAR(24),Q(24),W(5),IND(24),INDW(5)
DIMENSION G(256)
N=16
IFRA=(FRAME-1)*24
DO 1 KKK=1,4
  DO 1 KKI=1,6
    II=(KKK-1)*6*KKI
    IBLOCK=IFRA+II
    NUMBER=0
    READ(1*IBLOCK)G
175 CONTINUE
DO 2 K=1,16
  NUMBER=NUMBER+1
  VV(J,K)=G(NUMBER)
2 CONTINUE
175 CONTINUE
R1=0.0
YY(1,1)=0.0
DO 3 I=1,N
DO 4 K=1,N
R1 =R1+YY(I,K)
4 CONTINUE
3 CONTINUE
R31=R31+R1
R1=R1/255
WRITE(2,14)R1
14 FORMAT(1H ,F10.4)
R2=0.0
DO 10 I=1,N
DO 12 K=1,N
R2=R2+YY(I,K)**2
12 CONTINUE
10 CONTINUE
R32=R32+R2
R2=R2/255
R1=(R1)**2
VAR(I)=R2-R1
WRITE(2,33)I,VAR(I)
33 FORMAT(1HD,VAR OF BL',I5,'IS=',F10.5)
DO 32 I=1,24
Q(I)=VAR(I)
32 CONTINUE
IFAIL=0
CALL MO1AFK(Q,W,IND,INDW,24,5,FAIL)
IF(FAIL.NE.0)WRITE(2,100)FAIL
ICLASS=1
K=0
L=1
DO 24 I=1,24
WRITE(2,101)I,Q(I),INC(I)
101 FORMAT(1H ,I5,5X,F10.5,5X,I5)
   J=IND(I)
   IF(K.LT.6)GO TO 15
   ICLASS=ICLASS+1
   K=0
15  K=K+1
36  IF(J.LT.7)GO TO 25
   L=L+1
   J=J-6
   GO TO 36
25  MAP(L,J)=ICLASS
   L=1
24  CONTINUE
100  FORMAT(1H ,'SORT ERROR,IFAIL=',I3)
    WRITE(2,34)Q
34  FORMAT(12F10.5)
    WRITE(2,5)((MAP(I,J),J=1,6),I=1,4)
5   FORMAT(1H0,6I4)
    RETURN
END
FINISH
REFERENCES

1. ROSENFIELD, A., "Bandwidth Reduction Bibliography"
   I.E.E.E. Trans. Inf. Theory IT-14,4,
   pp. 601 - 602, July 1968

2. CLARKE, R.J., "Notes on Digital Coding of Visual
   Signals", Dept. of Electronic and Electrical
   Engineering, Loughborough University of
   Technology, U.K.

3. CLARKE, R.J., "On the Relation Between the Karhunen-
   Loeve and Cosine Transforms", I.E.E. Proc.,

4. PRATT, W.K., "Bibliography on Television Bandwidth

5. Special Issue on Digital Communications, I.E.E.E. Com.

   I.E.E.E., 63, 4, April 1975.

7. COCHRAN, et. al. W.T., "What is the Fast Fourier
   Transform?" I.E.E.E. Proc., 55, No. 10,

8. COOLEY, J.W., LEWIS, P.A., WELCH, P.D., "Historical
   Notes on the Fast Fourier Transform",
   I.E.E.E. Proc., 55, No. 10, pp.1675 - 1677,


78. GREGORY, R.L., "Eye and Brain, the Psychology of Seeing", Weidenfeld and Nicolson, 1966.


