	ig an de Star (1994) - 187				
					• •
			and galain an Angel Marine an Angel		
	BNDSC	no: DX 185	492	en e	• 
	L UNIVER	OUGHBOROUGH SITY OF TECHN LIBRARY	I IOLOGY		
	AUTHOR/FILING	TITLE	·····	• <u>•••</u> ••••••••••••••••••••••••••••••••	
	<u> </u>	HIA, L.T.	·	•	
	ACCESSION/COP	Y NO.			
	VOL. NO.	CLASS MARK		• • • • • • · ·	
		LOUND CORY			
	1 5 14 1950	- 3 OCT 1997	-		
	2 1 MAY 1996	2 5 JUN 1999			
	1 4 JUN 1996			1	
	1 4 .1111 1008			;	
an the state of the States of	27 JUN 1997			· · ·	
		an An Arasan an Arasan	• • • • •	•	



BADMINTON PRESS 18 THE HALFCROFT SYSTON LEICESTER LE7 BLD ENGLAND TEL: 0533:696636 FAX: 0533:696636

Ĵ

• . . ••• .

ζ.

.

# AN INVESTIGATION INTO THE REQUIREMENTS FOR AN EFFICIENT IMAGE TRANSMISSION SYSTEM OVER AN ATM NETWORK

Ţ

by

Liang Tien Chia B. Sc. (Hons)

**Supervisors** 

### Prof. J. W. R. Griffiths and Dr. D. J. Parish

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of the Loughborough University of Technology

© by Liang Tien Chia, May 1994

· · · · · · · · · · · · · · · · · · ·		
	Loughborough University	
	Date Der 94	
	Class	
	Acc. 040094275	
	v 8912143	
• a.		



This thesis looks into the problems arising in an image transmission system when transmitting over an ATM network. Two main areas were investigated. (i) An alternative coding technique to reduce the bit rate required and (ii) concealment of errors due to cell loss, with emphasis on processing in the transform domain of DCT-based images.

Initial work commenced with a study of image compression algorithms which lead onto an investigation of Fractals as an alternative to standard compression techniques. An implementation on a transputer-based system was investigated with enhanced modification to an initial fractal algorithm and results for monochrome images of compression ratios of 0.8 to 1.3 bits/pixel were achieved. Compression techniques, like transform coding, are capable of attaining similar compression ratios and an alternative fractal compression algorithm with much higher compression ratio was not identified.

Research then continued using standard compression algorithms but in order to maintain the resilience of the image transmission system, cell loss had to be compensated. Cells may be lost due to random bit error in a cell header or due to network control when traffic is congested and the treatment of video cell loss could be achieved with error correction or concealment techniques. This work concentrated on error concealment algorithms in the transform domain and many proposed algorithms were implemented and tested. Concealment techniques considered included replacing the lost blocks with a fixed colour block, replacement by the previous block within the same frame and replacement of the lost block with information derived from the neighbouring blocks. Emphasis was placed on the last method and statistical studies of the transform coefficients were conducted and the results used to improve the initial error concealment algorithms.

i

In particular, the beneficial mechanism of pre-selecting a subset of transform coefficients for regeneration of a lost cell was investigated. This reduction in the number of transform coefficients used for the algorithm also reduced the computation complexity. The ability to detect and conceal cell loss improved both the resilience of the image transmission system, and the quality of images after concealment were significantly better.



It gives great pleasure to express my deep gratitude to my supervisors, Professor J. W. R. Griffiths and Dr. D. J. Parish, for their guidance, encouragement and inspiration throughout this work. The invaluable suggestions and comments from both supervisors helped considerably to initiate and develop the work presented here.

The Author would like to acknowledge the financial assistance received from the Overseas Research Scholarship and Loughborough University of Technology, which made this research possible. Acknowledgement also goes to the Science and Engineering Research Council for equipments and software which were essential for this project.

Special thanks must also be extended to G.J.Lu, T.A.Rafik, I.W.Phillips, T.Rodgers, C.Nche and all my friends in the University and over the Internet for their help and useful discussion throughout this research period and for making this an enjoyable and memorable time.

My heartiest gratitude goes to my Mom and Dad and the rest of my family for their patience and understanding during my absence from home and for their constant encouragement. Especially my Mom and Dad, for bringing me into this world and providing me with the best and also for their financial support during my undergraduate days. My friends, who have kept me sane for the duration of my stay in the United Kingdom. To Sue, who taught me a lot and last but not least, L.K.Tan for her constant encouragements, support and belief in me.

> L.T. Chia (Clement) May 1994

iii



## **GLOSSARY OF ABBREVIATIONS AND SYMBOLS**

Rk	:- k-dimensional Euclidean Space.
$u_{i,j}, v_{m,n}$	:- 2-dimensional image array.
$\overline{R}, \overline{D}$	:- The mean values of the range and domain blocks respectively.
D	:- The contracted version of a domain block.
F(i,j)	:- 2-dimensional representation of DCT Transform Coefficients.
f(m,n)	:- 2-dimensional representation of a pixel block.
MSE, NMSE	:- Mean Square Error and Normalised Mean Square Error. The
	objective measurements used assessing images quality.
AAL	:- Asynchronous Transfer Mode Adaptation Layer.
ATM	:- Asynchronous Transfer Mode.
B-ISDN	:- Broadband ISDN (Integrated Services Digital Network).
BCH Codes	:- Error detection/correction method named after the researchers
	Bose, Chaudhuri and Hocquenghem.
CCITT	:- International Telegraph and Telephone Consultative
	Committee.
CIF, QCIF	:- Conformance Interchange Format (352×288 pixels), Quarter
	Conformance Interchange Format (176×144 pixels). Standard
	image size defined in the H.261 video conferencing standard.
CLP	:- Cell Loss Priority. A bit in the ATM cell header.
CPB	:- Constrained Parameters Bitstream. Defined in the MPEG
	format.
DC, AC	:- Transform coefficients, where DC is defined as $F(0,0)$ and the
Coefficients.	rest are known as AC coefficients.
DCT	:- Discrete Cosine Transform. A sub-optimum transform method.
DPCM	:- Differential Pulse Code Modulation.
FT	:- Fractal Transform Codes.
H.261	:- CCITT recommendation for n×64 kbits/s videoconferencing
	standard.

,

.

HVS	:- Human Visual System.
IDCT	:- Inverse Discrete Cosine Transform.
IFS	:- Iterated Function System.
ISDN	:- Integrated Services Digital Network.
ISO	:- International Standard Organisation.
ITU-TS	:- International Telecommunication Union - Teleconferencing
	Sector.
JBIG	:- Joint Bilevel Image Group.
JPEG	:- Joint Photographic Expert Group, the recommended
	compression technique for continuous-tone still images.
KLT	:- Karhunen-Loeve Transform.
NNI, UNI	:- Network Node Interface, User Node Interface. The two
	recommended type of ATM cell header.
MASTER	:- Multimedia AidS for TERconferencing. A SERC funded
	collaboration project between Loughborough University of
	Technology and Oxford Brooke University.
MATMX :	:- Model Asynchronous Transfer Mode eXchange. An exchange
	which has been developed from previous project collaboration.
MCU	:- Minimum Coded Unit.
M-JPEG	:- Motion JPEG (Joint Photographic Expert Group).
MPEG	:- Motion Picture Expert Group.
RGB	:- Red, Green and Blue Colour Space.
RIFF	:- Resources Interchange File Format.
QMF	:- Quadrature Mirror Filter.
SAR	:- Segmentation and Reassembly Layer.
SDH	:- Synchronous Digital Hierarchy.
VBR	:- Variable Bit Rate.
VLC	:- Variable Length Codes (entropy coding method).
VLSI	:- Very Large Scale Integration.
VQ	:- Vector Quantisation.
YUV, YC <sub>r</sub> C <sub>b</sub>	:- Luminance and Chrominance Colour Space.

Ŷ

.



# TABLE OF CONTENTS

ABSTRACT	Page i
ACKNOWLEDGEMENTS	111
GLOSSART OF ABBRE VIA HONS & STMBOLS TABLE OF CONTENTS	iv vi
1 INTRODUCTION	1
1.1 IMAGE TRANSMISSION SYSTEMS	1
1.2 TRANSMISSION MEDIA	2
1.3 VIDEO CELL STREAM	2
1.4 RESEARCH OBJECTIVES	3
1.5 THESIS ORGANISATION	3

1.5 THESIS ORGANISATION	
-------------------------	--

2	IMAGE COMPRESSION TECHNIQUES	6
	2.1 THE NEED FOR IMAGE COMPRESSION	6
	2.2 OVERVIEW OF IMAGE COMPRESSION TECHNIQUES	7
	2.2.1 DPCM and Predictive Coding	7
	2.2.2 Conditional Replenishment	8
	2.2.3 Motion Estimation	8
	2.2.4 Transform Coding	8
	2.2.5 Vector Quantisation	10
	2.2.6 Subband Coding	11
	2.2.7 Wavelet Transform/Packets Coding	12
	2.2.8 Model/Knowledge Based Coding	12
	2.2.9 Contour-Texture Oriented Techniques	13
	2.3 STANDARD IMAGE COMPRESSION ALGORITHMS	13
	2.3.1 JPEG Still Image Compression	13
	2.3.2 H.261 Video Codec for Audio-Visual Services	16
	2.3.3 MPEG Audio Visual System	18
	2.3.4 Other Developments in the Standard Bodies	21
	2.4 NOVEL IMAGE COMPRESSION USING IFS	21

4

•

.

3	IMAGE COMPRESSION USING ITERATED FUNCTIONS SYSTEM	23
	3.1 INTRODUCTION ON FRACTALS	23
	3.2 MATHEMATICS OF FRACTALS THEORY	24
	3.2.1 Transformations on Metric Spaces	24
	3.2.2 Transformations in the Euclidean Plane	24
	3.2.3 Contraction Mapping Theorem	25
	3.2.4 The Collage Theorem	25
	3.3 ITERATED FUNCTION SYSTEM	26
	3.4 BLOCK BASED FRACTAL CODING (PART I)	29
	3.4.1 Range Blocks	30
	3.4.2 Domain Blocks	30
	3.4.3 Searching of the Domain Pool	30
	3.4.4 Fast Decoding	31
	3.5 BLOCK BASED FRACTAL CODING (PART II)	32
	3.5.1 Domain Block Transformations	32
	3.5.2 Testing for a Block Match	34
	3.5.3 Block Matching in the Transform Domain	35
	3.5.4 Bits required for each Fractal Transform	37

Ş.

IMPLEMENTATION OF AN IFS SYSTEM	40
4.1 TRANSPUTER-BASED DISPLAY SYSTEM	40
4.1.1 Transputer Based Board (Controller)	41
4.1.2 Video Frame Store	41
4.1.3 Digital to Analogue Board	41
4.1.4 Utilisation of the Display System	42
4.2 PROGRAM FLOWCHARTS	42
4.2.1 Encoding of Monochrome Images	• 43
4.2.2 Decoding of Monochrome Images	45
4.3 RESULTS FROM THE SIMULATION	46
4.3.1 Test Results	46
4.3.2 Understanding the Result Sheet	47
4.3.3 Analysis of Results	47
4.4 SUMMARY AND CONCLUSION	49

5 TRANSMISSION OF PACKET VIDEO OVER AN ATM NETWORK	58
5.1 ASYNCHRONOUS TRANSFER MODE	58
5.2 CELL STREAM CONTROL OF PACKET VIDEO	60
5.3 TREATMENT OF VIDEO CELL LOSS	60
5.4 ERROR CORRECTION	61



6 ERROR CONCEALMENT ALGORITHMS	68
6.1 ERROR CONCEALMENT OF JPEG STILL IMAGES	68
6.1.1 Replacement by a Fixed Colour Block.	68
6.1.2 Replacement with Previous Macroblock in the Image	69
6.1.3 Replacement using Neighbouring Blocks	70
6.1.4 Interpolation Method	70
6.1.5 Replacement using Linear Interpolation of Neighbouring Blocks	71
6.1.6 Replacement using Quadratic Interpolation of Neighbouring Blocks	72
6.2 PERFORMANCE OF CONCEALMENT ALGORITHMS FOR	
JPEG IMAGES	73
6.2.1 Image 1: "Lena"	73
6.2.2 Image 2: "Peppers"	74
6.2.3 Image 3: "Toys"	74
6.3 ERROR CONCEALMENT OF M-JPEG IMAGE SEQUENCES	78
6.3.1 Replacement by Black Macroblock	78
6.3.2 Replacement by Previous Frame	78
6.3.3 Replacement by Previous Macroblock in the Same Frame	79
6.3.4 Replacement using Bilinear Interpolation of Neighbouring Blocks	79
6.4 PERFORMANCE OF CONCEALMENT ALGORITHMS FOR	
M-JPEG IMAGES	80
6.4.1 Image Sequence 1: "Movie"	80
6.4.2 Image Sequence 2: "Shoes"	81
6.4.3 Comparison between Methods	81
6.5 CONCLUSIONS AND FURTHER IMPROVEMENTS	82

7 STATISTICAL MEASUREMENTS OF IMAGES	91
7.1 TEST IMAGES USED FOR STATISTICAL MEASUREMENTS	91
7.2 DISTRIBUTIONS OF THE PIXELS AND TRANSFORM COEFFICIE	NTS 93
7.2.1 Histograms of Test Image A: "Lena"	93
7.2.2 Histograms of Test Image B: "Peppers"	96

7.2.3 Histograms of Test Image C: "Toys"	97
7.2.4 Histograms of Test Image D: "Random"	97
7.3 MODELLING OF THE TRANSFORM COEFFICIENT DISTRIBUTIONS	101
7.3.1 Linear Regression	101
7.3.2 Gaussian Distribution	102
7.3.3 Laplacian Distribution	108
7.4 CORRELATION OF TRANSFORM COEFFICIENTS	114
7.4.1 Correlation in Test Image A: "Lena"	116
7.4.2 Correlation in Test Image B: "Peppers"	122
7.4.3 Correlation in Test Image C: "Toys"	127
7.5 SUMMARY OF THE STATISTICAL MEASUREMENTS	129
7.5.1 Conclusion on Modelling of Distributions	129
7.5.2 1-Dimensional Correlation of DCT Images	132

÷.

8 MODIFICATION AND IMPLEMENTATION OF AN ERROR	
CONCEALMENT ALGORITHM	134
8.1 MODIFICATION TO THE ERROR CONCEALMENT ALGORITHM	134
8.2 RESULTS FROM THE SIMULATION	137
8.2.1 Comparison of Methods	137
8.2.2 Comparison of Methods (Magnified)	137
8.2.3 Conclusions and Discussions on the Modified Algorithm	140
8.3 IMPLEMENTATION OF THE CONCEALMENT ALGORITHM ON THE	
MASTER PROJECT	141
8.4 RESULTS OF THE IMPLEMENTATION	144

9	THE IMPORTANCE OF SPECIFIC TRANSFORM COEFFICIENTS	146
	9.1 CLASSIFICATION OF BLOCKS INTO SIMPLE SHAPES	147
	9.1.1 Four Basic Classes	147
	9.2 IMAGE IMPAIRMENT DUE TO THE LOSS OF SELECTED	
v	TRANSFORM COEFFICIENTS	149
	9.2.1 Human Perception System	150
	9.2.2 Estimation of Image Quality	151
	9.2.3 Test Images	151
	9.2.4 Subjective Assessment	154
	9.2.5 Objective Assessment	156
	9.2.6 Conclusions on Image Quality Assessment	158
	9.3 A SIMPLIFIED CLASSIFICATION	159
	9.4 CONCLUSIONS	160

10 FURTHER DEVELOPMENTS AND CONCLUSIONS	162
10.1 FURTHER DEVELOPMENTS	162
10.1.1 Fractal-based Image Coding	162
10.1.2 Additional Side Information	162
10.1.3 Adaptive Error Concealment Algorithms	163
10.1.4 Other Applications of Error Concealment Algorithms	163
10.2 SUMMARY AND CONCLUSIONS	163

REFÉRENCES	184
LIST OF PUBLICATIONS	182
APPENDIX D :-	
THE MASTER PROJECT	175
APPENDIX C :-	
STATISTICAL PROPERTIES	169
APPENDIX B :-	
DISCRETE COSINE TRANSFORM	167
APPENDIX A :-	

.

# CHAPTER 1



# CHAPTER 1: INTRODUCTION.

#### 1.1 IMAGE TRANSMISSION SYSTEMS.

The first offering of person to person image transmission system with real time, live moving images goes back some thirty years when the analogue picture-phone was exhibited at the 1964 World's Fair. In the last three decades, large companies have implemented videoconferencing systems that exist in expensive studios with true conferencing facilities, high quality sound and television quality video images. These image transmission systems are expensive, require a large bandwidth and are usually transmitted on leased lines.

In the late 80s, digital videoconferencing systems started to be widely available with the introduction of the H.320 videoconferencing standard and the associated H.261 video coding algorithm for  $n \times 64$  kbit/s [1]. Commercially available visual communication systems concentrated on providing large screen displays and most offer extra functionalities including graphics animation overlays, control of the camera at the remote end, chairman mode, etc. A few include proprietary compression algorithm to improve the performance when communicating with their own systems.

One of the key elements of a videoconferencing system is the video codec hardware which is required for reducing the amount of information to be transmitted. Advances in VLSI technology have reduced the size of the H.261 codec into one or two boards that sit in a desktop and the nation-wide availability of basic rate Integrated Services Digital Network (ISDN) as a viable transmission media has made it attractive to manufacture portable desktop systems. Cheap desktop videoconferencing systems are now appearing on the market with a price tag of around three to five thousand pounds.

Standard bodies have adopted a few other image compression techniques in the last few years. JPEG has been adopted for digital compression of still images [2] and MPEG is the recommended moving picture and audio standard for digital storage up to 1.5 Mbit/s [3]. The use of JPEG and MPEG image transmission systems is slowly being implemented in multimedia networks and the transmission of such images faces similar problems to videoconferencing systems.

#### **1.2 TRANSMISSION MEDIA.**

For image services to be popular, they needs to be available to the majority and to be available on the local area network (LAN). Most LANs are optimised for multiple users and data and are inherently unsuitable of handling real-time data. In a local environment, real-time data can only be transmitted if network utilisation is low and transmission delay is below the real-time constraint. The use of short, fixed cells in Asynchronous Transfer Mode (ATM) would be a good compromise for all the traffic types to be carried in a videoconferencing system and development of ATM-based LAN have been propelled by a group of vendors who formed the ATM Forum.

For communication over the wide area, the network will need to be configured to cope with voice, data, text and image transmission which have quite different communications profiles. The adoption of ATM as the switching protocol for implementing Broadband-ISDN will help to solve the problem of multimedia communication. An ATM-based public network should be available within the next two decades but most investments are presently in synchronous digital hierarchy (SDH) based transmission systems and the first implementation of ATM would most likely be carried over existing SDH networks.

#### **1.3 VIDEO CELL STREAM.**

The use of ATM opened the possibility of support to source coding of images by variable bit rate (VBR) with the concomitant advantages of consistent picture quality and bandwidth flexibility [4,5]. However, ATM capabilities are affected by delay jitters and cell loss which are important considerations for video applications. Cells may be lost due to random bit errors in a cell header or, probably more serious, to network control when traffic is congested.

In most cases an absolute guarantee that video cells will not be lost under any circumstances cannot be given, and the video service must accept this possibility. Such a possibility must be considered at the video design stage, and may require the development of more robust video coding schemes at the expense of maximising video compression.

#### **1.4 RESEARCH OBJECTIVES.**

The research described in this thesis looks into the problems faced by an image transmission system over a packet network, specially over an ATM network. Two main issues which this work is concerned with are;

- The search for alternative image compression algorithms so as to reduce the bandwidth requirements.
- The improvement in the resilience of existing image compression algorithms by the introduction of error concealment methods to correct for cell loss in an ATM network.

The research was carried out along the following lines to achieve the objectives set out above;

- To study standard image compression algorithms and to search for alternative compression algorithms.
- An investigation into an alternative image compression algorithm and to justify its suitability as the preferred video compression method.
- To look into the problem of cell loss due to random bit error or traffic congestion which is inevitable in a packet network.
- To conduct an elaborate study on possible error concealment techniques and to implement algorithms for concealment of still images and image sequences affected by transmission errors.

#### 1.5 THESIS ORGANISATION.

**Chapter 1**: Gives an introduction to the work reported in this thesis. Research objectives are stated and an outline of the thesis is given in this chapter.



**Chapter 2:** The need for compression is shown and an overview of image compression techniques are presented in this chapter. The chapter also gives a brief description of standard compression algorithms like JPEG, MPEG and H.261. The last section describes a novel image compression technique using an Iterated Function System.

**Chapter 3**: This chapter introduces fractals, some associated mathematics and the use of Iterated Function System in image compression. The main part of the chapter then goes on to describe the implementation of a modified block based fractal coding algorithm.

**Chapter 4 :** An Occam decoding program was written and implemented on a transputer based display system. The encoding program was also implemented on a transputer. The algorithm is asymmetrical and encoding is computationally much more intensive than the decoding process. Results from the simulation are presented and conclusions are drawn about the compression algorithm.

**Chapter 5**: A brief description of the Asynchronous Transfer Mode of transmission is given. Transmission of packet video over an ATM network requires some sort of cell stream control and treatment of video cell loss. A decision was taken to concentrate on standard DCT-based algorithms and to look into error correction and error concealment techniques. In particular, the work focuses on error concealment algorithms for JPEG and M-JPEG images. The last section of this chapter describes the JPEG and M-JPEG frame structure.

**Chapter 6:** A number of error concealment algorithms were tested and described in this chapter. The list of concealment algorithms includes replacement by a fixed colour block, replacement by the previous macroblock in the same image, replacement by the previous frame or macroblocks from the previous frame (for correlated image sequences), replacement with information derived by linear interpolation of neighbouring blocks and finally replacement with information from quadratic interpolation of neighbouring blocks. The performances of the algorithms were tested on still JPEG images and M-JPEG image sequences. This chapter ends with conclusions drawn from results of the concealment algorithms and suggests possible further improvements.

**Chapter 7:** Statistical measurements of images are described in this chapter. Four images were used and the distribution of the pixel, DC and first few AC



**Chapter 8**: The error concealment method using linear interpolation of neighbouring blocks provided the best results. Based on an understanding of the statistics in the images, the algorithm is modified by using only a subset of the transform coefficients. This reduced the computational complexity and achieved comparable results. The algorithm was tested on the MASTER Project (as described in Appendix C) and the objective of this was to see if the algorithm could be implemented in real time.

**Chapter 9 :** This chapter starts with an assessment of image impairment due to the omission of selected transform coefficients. Subjective and objective measurements were taken and the results emphasise the importance of selected coefficients. The next section concentrates on the classification of transform blocks into a few classes of simple shapes. The reason for conducting this work is to lay the initial foundation for future developments.

*Chapter 10*: The final chapter provides suggestions for further developments and a summary and conclusion of the research work done.

Appendices : Appendix A is on the Discrete Cosine Transform and this provides the 2-dimensional DCT and IDCT equations with the matrix representations. Appendix B describes the statistical properties of a first order autoregressive process, the equation for calculating correlation coefficients and a section which interprets the correlogram, which is a plot of the correlation coefficients against the lag. Appendix C contains a brief description of the MASTER Project, a SERC funded project to investigate a low cost teleconferencing environment suitable for operation across an Integrated Services Digital Network (ISDN). Appendix D is a list of publications by the author.

# CHAPTER 2 ▷▷

.



# CHAPTER 2 : IMAGE COMPRESSION TECHNIQUES.

#### 2.1 THE NEED FOR IMAGE COMPRESSION.

To transmit moving digital colour images, even of relatively low definition, requires a very large amount of data unless it is processed in some manner to take advantage of the redundancies in the image. For example, a full colour image of  $512 \times 512$  pixels of 8 bits for each of the three primary values requires 6 Mbits per frame and for a frame rate of 25 frames/sec, this would translate into a transmission rate of 150 Mbit/s. Even a monochrome image of  $512 \times 512$  pixels with 8 bits for pixel intensity requires a storage capacity of 2 Mbits (256 Kbytes) per frame, therefore a high density  $5\frac{1}{4}$ " floppy disk can only accommodate 4 uncompressed monochrome images or 1 uncompressed 24 bit colour image. Fortunately the physiology of the human visual system together with the nature of the images allows very significant reduction in this rate.

With the continuing growth of modern communications technology and advances in computer technology for mass storage and digital processing, the demand for image transmission and storage is increasing rapidly. These have paved the way for the implementation of advanced data compression techniques to improve the efficiency of transmission and storage of images. A thorough review of early image compression techniques can be found in "Picture Coding: A Review" - by Arun N. Netravali, John O. Limb [6] and "Image Data Compression: A Review" by Anil K Jain" [7]. Recent reviews on new compression techniques includes "Second-Generation Image-Coding Techniques", by M.Kunt, M.Kocher, A.Ikonomopoulos [8] and "Image Coding - From Waveforms to Animation" by R.Forchheimer, T.Kronander [9].



#### 2.2 OVERVIEW OF IMAGE COMPRESSION TECHNIQUES.

Image compression techniques have been actively researched for the past forty years and this section will concentrate both on spatial and temporal techniques, some mature and others relatively new. The goal of image coding is to reduce, as much as possible, the number of bits necessary to represent and reconstruct a faithful duplicate of the original picture. This is a very reasonable goal as data originating from an image are not random and early methods explore this redundancy in the data. For example, a complete image of constant grey level is fully predictable once the grey level of the first pixel is known. Most compression algorithms therefore attempt to represent a given sampled image array  $\{u_{j,k}\}$  by another array  $\{v_{m,n}\}$ , which has little or no redundancy and to use this array to produce a near replica of the original sampled image.

#### 2.2.1 DPCM and Predictive Coding.

DPCM is a common data transmission method utilising predictive quantisation. The underlying principle behind DPCM is the intersample dependence of the sequence of samples. A causal prediction is formed from nearby, already processed, pixels of the present element to be coded. The difference, between the predicted and actual pixel, is then efficiently quantised and coded. The compression ability thus depends on the variance of the sampled images.

Another method is Delta Modulation which is the simplest form of predictive coding. A one bit representation of the signal is achieved by comparing the previous sample and the present one. The primary limitations are slope overload when there is a large jump or discontinuity in the signal and granularity noise which is due to the step-like nature of the output signal when the input signal is almost constant. An adaptive predictive-coding technique can be extended to image sequences with a switched predictor operating on the basis of local image activity.

#### 2.2.2 Conditional Replenishment.

Conditional replenishment is a simple method of detection and coding of the moving areas from frame to frame. The magnitude of the sampled data is compared with the magnitude of the previous sampled data, only if the difference is greater than a predetermined threshold would the difference be sent. At the receiver, a pixel is reconstructed either by repeating the value of the previous sample if it is stationary or by replenishing it by the decoded difference signal.

#### 2.2.3 Motion Estimation.

Motion compensation adaptively varies the predictor, usually according to an estimate of the translational motion of objects. If the estimate is accurate, the prediction is improved by use of appropriately spatially displaced pixels from the previous frame. A high degree of accurate motion compensation is still not

achievable and present implementation relatively uses simple block based techniques. Each block picture is slid along neighbouring blocks in the previous frame to find the "best match" in the search area. A full search is optimum but the computational demands are high. A commonly used suboptimum search algorithm is shown in figure 2.1. In this search only three iterations would be required for an area of  $15 \times 15$ pixels.



#### 2.2.4 Transform Coding.

Transform coding achieves compression by an energy preserving (orthogonal) transformation of the given image into another array such that the maximum information is packed into a minimum number of samples [10]. As shown in Figure 2.2a and 2.2b, the picture is divided into blocks and then each of these blocks is transformed into a set of more independent coefficients. The coefficients after transformation are then quantised and coded for transmission.





At the receiver, a reverse process takes place, the received bits are decoded into transform coefficients and an inverse transformation is applied to the transform coefficients to recover the pixel intensity of the blocks.

#### Optimum Transform (Karhunen-Loeve Transform).

The optimum transform will results in the best picture quality using the least number of bits. This transformation would produce uncorrelated coefficients and the image energy after transformation would be compacted into as few coefficients as possible. Due to quantisation and the dropping of specific coefficients in the encoding process, the mean-square error after reconstruction should be minimised for the optimum transform.

Fortunately, there is a transform that satisfies these criteria, and it is known as the Karhunen-Loeve Transform (KLT). However, practical implementation of the KLT involves estimating the auto-variance matrix of the data sequence and determination of its eigenvectors (basis vectors) and eigenvalues (coefficients) which requires a lot of processing. The need to do this calculation for each frame has made KLT, although ideal, an impractical tool. Therefore, in practice, the KLT is usually substituted by the sub-optimum unitary transforms.



Many other Transforms have been invented which produce less correlated coefficients than the original image itself and which are easier to implement then the KLT. The basic principles are similar. What is required for a two dimensional transformation is shown below :

$$V = A U A^{-1}$$
  $U = A^{-1} V A$  ... (2.1)

where  $U = u_{k,l}$  and  $V = v_{m,n}$  are the matrix notations of the 2-dimensional input and transformed image respectively.  $A = a_{k,l} \& A^{-1} = A^{T} = a_{l,k}$  are the matrix notations for the unitary and inverse unitary transformation respectively. k, l, m, nare the size of the matrix.

Some popular sub-optimum transforms are the 2-dimensional Fourier Transform, the Discrete Cosine Transform, the Discrete Sine Transform and the Hadamard Transform. A more detailed description of the Discrete Cosine Transform (DCT) is in appendix A.

#### 2.2.5 Vector Quantisation (VQ).

An often quoted result of Shannon's rate distortion theory is that more efficient coding can always be achieved by processing vectors, rather than scalars. The basic idea is simple and was initially used for images in the early 1980s. A vector quantiser can be defined as a mapping Q of the k-dimensional Euclidean space  $\Re^k$  into a finite subset Y of  $\Re^k$ . Thus,

$$Q: \mathfrak{R}^k \to Y \qquad \dots (2.2)$$

where  $Y = \{\hat{x}_i; i = 1, 2, ..., N\}$  is the set of codebook entries and N is the total number of entries in the codebook.

An image is divided into small blocks and each is, in turn, compared with similar entries in a codebook. For the closest match an index,  $\hat{x}_i$ , corresponding to that entry is transmitted to the receiver (which has access to the codebook) which then uses the same codebook entry for reconstruction. Codebook design is traditionally carried out using the so called LGB algorithm [11], based on a clustering approach. Efficient codebook search algorithms have been intensively researched over the last 10 years with various sub-optimal techniques being proposed.[12]



#### 2.2.6 Subband Coding.

The reason for using subband techniques for coding is that the subbands are more easily encoded than the original signal, i.e., the subbands should be uncorrelated both within each subband (autocorrelation) and between different bands (cross correlation). Other advantages of subband coding include the absence of blocking artefacts and the resilience to loss in packet networks.

As in transform coding, the subband coder aims at dividing the image signal into uncorrelated parts, but rather than use a block transform, the decomposition is made through filtering.

Multi-dimensional quadrature mirror filter (QMF) techniques and algorithms have been reported by Vetterli [13] and implemented in various subband coding algorithms[14,15] since 1986. An image is split into a number of separate frequency bands and possible two-dimensional decomposition structures for 4 and 7 bands are shown in figure 2.3. Aliasing cannot be completely avoided but can be controlled by proper filter design, such as the use of symmetric linear phase filterbanks.

After decomposition, the subband signals can be coded by conventional coding techniques including vector quantisation, discrete cosine transform, adaptive/non-adaptive memoryless quantisation or predictive coding. Except for the lowest band, the spatial correlation between elements in the upper subbands is low. Therefore, it is sensible to use statistical methods like variable and runlength coding for the encoding of upper subbands.





#### 2.2.7 Wavelet Transform/Packets Coding.

The class of wavelet techniques is not really well defined and it keeps changing. Hence, it is virtually impossible to give a precise definition of "wavelet" that incorporates all different aspects. Primarily, wavelets are families of functions generated by dilating and translating a unique, zero mean, generating function known as the mother wavelet.

$$\Psi^{a,b}(t) = |a|^{-\frac{1}{2}} \Psi\left(\frac{t-b}{a}\right) \qquad \dots (2.3)$$

where a is the dilation factor, b is the translation

High frequency wavelets correspond to a < 1 or narrow width, while low frequency wavelets have a > 1 or wider width. The time-frequency relationship of a wavelet is shown in figure 2.4. In comparison, a window Fourier transform has fixed resolution in both the spatial



and frequency domains, whereas a wavelet transform varies with the dilation parameter. The wavelet transform can be shown to decompose the signal into a set of frequency bands having a constant size on a logarithmic scale [16]. There is a one-to-one correspondence between orthonormal wavelet bases and discrete multiresolution analysis, which implies a strong relationship between subband coding and wavelet transform. Wavelets can therefore be used for decomposition of the image into its components, each wavelet picks up information about the image at a given location and a given scale. Wavelet coefficients can be encoded using conventional coding algorithms and reconstruction can be achieved from the coefficients transmitted and knowledge of the wavelet function. [17,18]

#### 2.2.8 Model/Knowledge Based Coding.

Knowledge-based image coding utilises general knowledge about the coded subjects as well as 3-dimensional shape model [19]. The encoder analyses the input images, transmits the relevant parameters and the decoder synthesises output images based on the knowledge transmitted from the encoder. Very low bit rate can be achieved with this method but the major problems are the difficulty in modelling unknown objects and the presence of analysis errors. A large proportion of research in this area concentrates on analysis and synthesis of facial expressions. The facial model is well studied but to generalise the application, it is necessary to reconstruct a 3-D structure of the scene from 2-D images and reconstruction of unmodelled objects are proving to be difficult. Research in other image processing areas such as feature extraction, object recognition and the use of artificial intelligence will contribute to the development of knowledge based coding.

#### 2.2.9 Contour - Texture Oriented Techniques.

Contour-texture oriented techniques attempt to segment the image into textured regions surrounded by contours such that the contours correspond, as much as possible, to those of the objects in the image. Contour and texture information are coded separately. Contours may be extracted either by the method of region growing or edge detection techniques. In the first case, closed contours are obtained which make it easy to list regions and their properties. In the latter, contours obtained are not necessarily closed. After removing the contour information, there is no longer any sharp discontinuity within each region, hence the variation of the pixel level within a region can be described with smooth two-dimensional polynomial functions.[8]

A recent paper [9] on Contour-texture technique extracts contour information from the local maxima of the wavelet transformation and textures are coded within a wavelet orthonormal basis.

#### 2.3 STANDARD IMAGE COMPRESSION ALGORITHMS.

For digital image applications involving storage or transmission to become widespread in today's marketplace, a *standard* image compression method is needed to enable interoperability of equipment from different manufacturers. A few standards have emerged in the last couple of years and a brief introduction of each is given below.

#### 2.3.1 JPEG Still Image Compression.

A standardisation group known by the acronym JPEG, for Joint Photographic Experts Group, has been working towards establishing an international digital image compression standard for continuous-tone (multi-level) still images, both greyscale and colour.



The Joint in JPEG refers to a collaboration between the ITU-TS (formerly the CCITT) and the ISO. The JPEG has undertaken the ambitious task of developing a general purpose compression standard to meet the needs of almost all continuous tone still image applications [20,21].

Block diagrams of the proposed JPEG Encoder and Decoder are shown in figure 2.5a and 2.5b. A brief description of the standard is presented here, the first stage of the JPEG algorithm being Discrete Cosine Transform (DCT). DCT transforms image data from the spatial to the frequency domain. The transform is achieved by dividing the image into  $8 \times 8$  blocks and applying a two-dimensional DCT to each block. If an image is not a multiple of 8 pixels wide or high, the images are padded to the next 8 pixel boundary.

The second stage of the JPEG algorithm quantises each frequency component from the DCT transform. The quantisation factor for each coefficient is stored in an internal table and the factors are usually based on the Human Visual System. Typically, DC component and the low-frequency components use lower quantisation factors, and the high frequency components use the high quantisation factors.

A typical set of input pixel values, the corresponding set of coefficient values and the values after quantisation are shown in figure 2.6. The transform coefficients after DCT are shown in fig 2.6b and fig 2.6c, before and after quantisation respectively. After quantisation, based on the JPEG recommended values, it can be seen that 48 out of 64 coefficients are zero, with the non-zero values concentrated in the top left hand corner.

Varying amounts of detail in the source block will produce different sets of coefficients. The example shown in Figure 2.6 is for a quite complex block, whereas a more even area would produce far fewer coefficients. A uniform area, say of the sky, after quantisation could quite possibly result in just a single DC coefficient remaining. The coefficients are then rearranged according to the zigzag scan order.

The third stage of the algorithm encodes the output of the quantiser. For the DC component, the encoder subtracts the DC element of the previous block from the

									0.00							
129	101	71	49	40	35	44	49		869	-231	-27	30	-35	8	4	-10
137	120	92	59	48	41	51	53		186	89	-83	-6	25	-17	7	-5
139	134	119	87	62	48	47	54		9	63	56	-20	-11	6	-6	3
144	149	147	118	93	60	49	44		-1	-13	23	13	-2	-4	1	-1
146	156	158	142	123	94	63	49		-7	6	5	0	5	3	0	2
153	163	165	159	159	139	122	96		0	-4	7	-1	-2	4	•2	1
144	137	144	144	148	153	156	142		-2	1	•2	-2	1	1	-2	4
141	135	131	133	137	140	139	127		9	1	-1	1	-2	2	-2	-4
(a)	A set	t of ty	pical	inpu	t pixe	l valı	les	-	(b) ]	The c	orres	ondi	ng co	effici	ient v	alu
			-		_								-			
54	-21	-3	2	1	0	0	0		-		~ •	7		-,	<i>,</i> —	
54 16	-21 7	-3 -6	2	1	0	0	0				, , / , /	7			. — / `	
54 16 1	-21 7 5	-3 -6 -4	2 0 -1	1 1 0	0 0 0	0 0 0	0 0 0		- - -		/ · / /	7			. <del>-</del> / ' /	
54 16 1 0	-21 7 5 -1	-3 -6 -4 1	2 0 -1 0	1 1 0 0	0 0 0 0 0	0 0 0	0 0 0				/ · ·	7 / /		-, , , , ,	. <del>-</del> / , /	
54 16 1 0	-21 7 5 -1 0	-3 -6 -4 1 0	2 0 -1 0	1 1 0 0	0 0 0 0	0 0 0 0	0 0 0 0					7			 / / / / /	
54 16 1 0 0	-21 7 5 -1 0 0	-3 -6 -4 1 0 0	2 0 -1 0 0 0	1 1 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0					7				
54 16 1 0 0 0 0	-21 7 5 -1 0 0 0	-3 -6 -4 1 0 0	2 0 -1 0 0 0 0	1 1 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0					7				

current block and encodes the difference using a Huffman coding technique. For the AC components, the encoder first groups together any consecutive zeros before a non-zero AC element, and then encodes the run-length of the encoded zeros with the AC element using a Huffman coding technique.

Decompression uses the same modules as compression but in the reverse direction. The incoming data is passed through a run-length decoder to generate the DCT coefficients for the Inverse Discrete Cosine Transform (IDCT) which transforms the coefficients back into  $8 \times 8$  pixel blocks.

#### 2.3.2 H.261 Video Codec for Audio-Visual Services.

The ITU-TS (International Telecommunications Union - Teleconferencing Sector) recognising the need for the provision of providing widespread videoconferencing services using Integrated Services Digital Network (ISDN), established a specialist study group XV, in 1984, to look into coding for Visual Telephony, with the objective of recommending a video coding standard for transmission at  $m \times 384$  kbit/s (m = 1, 2, ..., 5)

Later in the study period, with new discoveries in video coding techniques, it became clear that a single standard,  $p \times 64$  kbit/s (p = 1, 2, ..., 30), can cover the entire ISDN channel capacity. After more than five years of intensive deliberation, the ITU-TS recommendation H.261, Video Codec for Audiovisual Services at  $p \times 64$  kbit/s was completed and approved in December 1990. The intended applications of this international standard are videophones and video conferencing. Therefore, the recommended video coding algorithm has to be able to operate in real time with minimum delay. For p = 1 or 2, due to the severely



limited bitrate, only desktop face-to-face communication (videophone) is appropriate. For  $p \ge 6$ , due to the additional bandwidth, more complex pictures can be transmitted with better quality. This is more suitable for videoconferencing.

Figures 2.7a and 2.7b shows the block diagrams of the CCITT H.261 Encoder and Decoder respectively. The CCITT Encoder is based on transform coding and temporal predictive coding with motion compression. Each macroblock consists of four  $8 \times 8$  luminance blocks and two  $8 \times 8$  chrominance blocks. (based on CCIR 601 4:2:2 standard). The chrominance blocks have reduced resolution because the human visual system is less sensitive to colour detail than luminance detail.

should be d And to g T

14:2:0

The encoder has two modes of operations. The intra-frame mode realises compression within one frame but only in the spatial dimension. Compression in the temporal direction is disabled. This mode is used for coding the first frame in a scene and for resetting the prediction loop. The inter-frame mode, in addition to compression within one frame, also realises compression between consecutive frames. The previous decoded frame is motion compensated and then used as a prediction for the current frame.

Motion compensation is performed on  $16 \times 16$  luminance blocks within each macroblock with a maximum displacement of 15 pixels allowed in each dimension. A loop filter is normally activated to remove the blocking artefacts associated with motion compensation (as disjoint blocks of pixels are joined, artificial edges may be created).







Zigzag scanning is used to arrange the quantised DCT coefficients in order of ascending frequencies. As high frequency coefficients are most likely to be zero, this ordering result in the longest runs of zeros. The events are variable length coded, combined with the header information (quantisation stepsize, inter-intra mode, motion vectors, etc.) and then buffered. Commonly occurring events are coded with as few as two bits while rarely occurring events are coded with up to 20 bits. Synchronisation and check bits are also added to the bit stream to allow error correction in the decoder. The error correction code is BCH(511,493) which can correct up to 2 bit errors in every block of 511 bits. This format contains 18 redundant bits for each 493 data bits.

In the H.261 decoder, the compressed data arriving is buffered and then variable length decoded. To reconstruct the decoded frame, the quantised coefficients are processed using an Inverse Discrete Cosine Transform (IDCT). The output of the IDCT represents the reconstructed prediction error (or the reconstructed current frame itself in the intra-frame mode). This output is added to the current frame prediction (or zero in the intra-frame mode) to obtain the decoded frame which is then stored in a frame memory.

#### 2.3.3 MPEG Audio Visual System.

Another standardisation effort is known by the acronym MPEG (Moving Picture Experts Group) and is part of the ISO-IEC/JTC1/SC2/WG11 working group. The MPEG's activities are divided into three parts and they cover more than video





compression, since the compression of the associated audio and the issue of audio-visual synchronisation cannot be worked independently of the video compression; Part 1: MPEG-Audio looks at compression of a digital audio signal at rates of 64, 128 and 192 kbit/s per channel, Part 2: MPEG-Video is addressing the compression of video signals at about 1.5 Mbits and Part 3: MPEG-System addresses the issue of synchronisation and multiplexing of multiple compressed audio and video bit streams. The activity of the MPEG committee was started in 1988 and an updated committee draft was produced in March 1992.

MPEG Video: Part 2, is described in CD 11172-2 "Coding of Moving Pictures and Associated Audio - for digital storage media at up to about 1.5 Mbit/s" and an outline is presented below with the block diagrams of the MPEG encoder and decoder shown in figure 2.8.

The intention of the International Standard has been to define a source coding algorithm with a large degree of flexibility that can be used in many different applications. A subset of the Standard has been defined, the Constrained Parameters Bitstream (CPB) are a set of sampling and bitrate parameters designed to normalise computational complexity, buffer size and memory bandwidth. CPB limits video to 396 macroblocks (101 376 pixels) per frame and a maximum bitrate of 1.856 Mbit/s, which is suitable for digital storage media such as CDs, Digital Audio Tapes (DAT), magnetic and optical drives and allows cost effective VLSI implementations in 1992 technology (0.8 microns).

The standard needs to balance a high picture quality and compression ratio with the requirement for random access to the coded bitstream. To obtain good picture quality at the bitrates of interest would require a very high compression ratio, which is not achievable with present intraframe coding alone. The need for random access in the bitstream is best satisfied by pure intraframe coding. Therefore a balance between inter and intra-frame coding is necessary.

#### Temporal Processing.

Three main picture types are defined. Intra coded pictures (I-Frames) are compressed with (DCT) transform domain based compression for the reduction of spatial redundancy without reference to other pictures and they provide access points to the coded picture sequence where decoding can start. Predictive coded pictures (P Frames) are coded more efficiently with motion compensated prediction from a past intra coded or a predictively coded picture and this will be used as a reference for further prediction. Bidirectionally predictive coded pictures (B-Frames) provide the highest level of compression with motion compensated prediction from both past and future reference pictures.

Bidirectionally predictive coded pictures are never used as references for prediction. The numbers and organisation of the three different picture types in a sequence are very flexible and figure 2.9 illustrates a typical relationship.



#### MPEG Video Encoding.

The standard requires the input video signal to be digitised and represented as one luminance and two colour difference signals  $(YC_rC_b)$ . The colour difference signals are subsampled with respect to the luminance by 2:1 in both the vertical and horizontal directions. The basic unit of coding within a picture is  $16 \times 16$ macroblocks, each macroblock consisting of six  $8 \times 8$  blocks, 4 luminance and two chrominance blocks. For I-Frames, DCT is used as the block based compression technique with visually weighted quantisation, variable and runlength coding is then used to encode the coefficients efficiently. Predictive coded macroblocks in the P-Frames may either contain motion vectors or not pending the outcome from the motion compensation criterion. If the motion vector is zero, the macroblock is not transmitted. The prediction error, if necessary, is further compressed with spatial redundancy deduction (DCT) and transmitted with the motion information. Macroblocks in B-Frames could be predicted from forward, backward and interpolative compensation modes and the residual errors, if large enough, could be compressed further using the DCT transform. All information is compressed further using variable and run-length codes to achieve maximum


efficiency. Before writing to the bitstream, the pictures have to be arranged into the order in which the decoder processes them.

#### MPEG Video Decoding.

Decoding is the inverse of the encoder operation. It is considerably simpler than encoding as there is no need for motion estimation and there are fewer options. As the decoder reads the bitstream, it identifies the start of a coded picture and the type of the picture. Each macroblock in the picture is then decoded in turn. After all the marcoblocks in the picture have been processed, the picture can be reconstructed. An I or P Frame is stored as a reference frame for subsequent pictures. Before the pictures are displayed, they need to be reordered from the coding order to their natural display order.

#### 2.3.4 Other Developments in the Standard Bodies.

The ITU-TS is defining an improvement to H.261 with the participation of industry vendors. The collaboration between the ITU-TS and the ISO are looking into the JBIG (Joint Bilevel Image Group) compression standard, which consists of lossy and lossless compression techniques for bilevel images. The JPEG-2 compression standard will be an upgrade and includes adaptive quantisation within an image.

Other groups in the collaboration are concerned with further developments of moving picture and associated audio for digital storage. MPEG was optimised for CD-ROM or applications at about 1.5 Mbit/s whereas MPEG 2 addresses applications at broadcast TV sample rates using the CCIR 601 specifications (720 samples/line  $\times$  480 lines per frame  $\times$  30 frames per second or about 15.2 million samples/sec including chroma) as reference and at a bitrate of about 15 Mbit/s.

MPEG 4 targets the very low bitrate applications defined loosely as having sampling dimensions up to  $176 \times 144$  (QCIF)  $\times 10$  Hz and coded bit rates between 4.8 - 64 Kbit/s.

### 2.4 NOVEL IMAGE COMPRESSION USING IFS. (ITERATED FUNCTION SYSTEM)

One particularly imaginative compression technique which has gained a lot of interest and controversy is the use of fractals. Initial claims in the popular press quoted image compression ratios of up to a million to one but the professional literature only reported compression ratios comparable with other modern compression techniques. This technique was developed by a mathematician [22] working in the US, Dr. Michael Barnsley - until recently a Professor of mathematics, now one of the co-partners in the firm Iterated Systems Ltd. Fractal patterns are complex but have a low information content, so can be specified with small amounts of data. This technique is based on an aspect of fractal mathematics called Iterated Functions Systems. These are sets of mathematical functions that describe a sequence of transformations of an image. When applied repeatedly, they generate progressively smaller images that eventually converge on a unique fractal. Barnsley's system finds collections of transforming functions that uniquely define any image, and present these as Fractal Transform (FT) codes. A large amount of computation is required for encoding. Equally important is the fact that it can then recover the image from the fractals. The functions can be expressed in a standard form, so that the FT codes represent the key constants in the function set, allowing very efficient representation.

The company have produced the P.OEM product line which consists of the compression board for Fractal Transform technique compression and storing of video images. Decompression occurs in software, as part of the application program and no special hardware is required. As specified in the commercial brochure, colour still image with an image panel of  $320 \times 200$  pixels and 24 bits colour can be compressed into a file size of 6 Kbytes. The original file size would be 192 Kbytes, thus achieving a compression ratio of 38:1. With a 25 MHz 80386 computer, it takes less than one second to read, decompress and display a full colour image.

# CHAPTER 3

\*

•

## CHAPTER 3 : IMAGE COMPRESSION USING ITERATED FUNCTION SYSTEM.

#### 3.1 INTRODUCTION ON FRACTALS.

Fractal is a word initially coined by B.B. Mandelbrot [23] from the Latin *fractus*, meaning broken, to describe objects that were too irregular to fit into a traditional geometrical setting. Very attractive images of the Mandelbrot and Julia sets can be generated from a simple equation  $z_{n+1} = z_n^2 + c$  where z and c are complex numbers. Each point on the complex plane is applied to the equation and the number of iterations required to reach an agreed maximum value of this function, usually near infinity, is recorded. Plotting the graph in the complex plane with different colours to represent the different range of iterations will generate the colourful images shown in numerous colourful pictures and brochures.

Numerous applications have since been found for fractals and the related study of Chaos Theory. Applications include examination of long-range fractal correlations for DNA base sequences[24], analysis and synthesis of texture with fractals [25] and a study of possible chaotic behaviour in the stock market [26]. In the last few years, great interest has been generated in fractal representation of images [27].

During the 80s, a few mathematicians at the Georgia Institute of Technology including M.F. Barnsley started addressing the problem of applying IFS to generate two- and three-dimensional objects. This work concerns the development of IFS as a practical tool for the reproduction of images including clouds, smoke, horizons, seascapes, flames and branching structures such as leaves and ferns. For images with objects like those mentioned above, which exhibit deterministic self similarity, IFS would be an ideal tool for representing the image. Reports of this method in the popular technical press generated



enormous interest in this subject. Compression ratios of 1000-10000 to 1 are possible for images with recognisable objects that exhibit a large degree of self similarity.

For every object in the image a set of contractive affine transformations, collectively known as the IFS, can be found to represent that object. For an image that consists of a number of different objects, the principle is applied to each object and the collective sets of affine transformations can be used to describe the picture. Encoding is computationally very intensive and identifying objects is difficult. Images generated from this method look similar to a computer graphics constructions which superficially, appear somewhat like natural objects.

#### 3.2 MATHEMATICS OF FRACTALS THEORY.

#### 3.2.1 Transformations on Metric Spaces.

Fractal geometry studies subsets of geometric spaces, the focus being on subsets of a space which are generated by simple geometrical transformations of the space into itself. Simple transformations can usually be specified by a small set of parameters. Let (X,d) be a metric space and a transformation on X is a function  $f: X \to X$  which assigns exactly one point  $f(x) \in X$  to each point  $x \in X$ . Examples include affine transformations in  $\Re^2$  space, which can be expressed using  $2 \times 2$  matrices and  $2 \times 1$  column vectors.

To work in fractal geometry one needs to know well the relationship between formulae for transformations and geometric changes, such as stretching, twisting, folding, and skewing of the underlying fabric and of the metric space upon which they act on individual points.

### 3.2.2 Transformations in the Euclidean Plane ( $\Re^2$ space).

A transformation  $W: \mathfrak{R}^2 \to \mathfrak{R}^2$  of the form

$$W\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \qquad \dots (3.1)$$
  
or  
$$W(x) = Ax + b$$

where a,b,c,d,e and f are real numbers, is called a (two dimensional) affine transformation. Here  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  is a two dimensional 2 x 2 real matrix and **b** is

the column vector  $\binom{e}{f}$ . The general affine transformation W(x) = Ax + b in  $\Re^2$  consists of a linear transformation, A, which deforms space relative to the origin as described above, followed by a *translation* or *shift* specified by the vector b.



#### 3.2.3 Contraction Mapping Theorem.

A Transformation  $f: X \to X$  on a metric space (X,d) is called contractive or a contraction mapping if there is exactly one fixed point  $x_f \in X$ . For any point x, on the complex space, the transformation converges to the fixed point,  $x_f$ . Figure 3.2 illustrate the idea of a contraction transformation on a compact metric space.



#### 3.2.4 The Collage Theorem.

The Collage theorem provides a systematic method for finding the affine transformations that will produce an IFS encoding of a desired image. A target image T is defined (for example, a black leaf on a white background). A contractive affine transformation is initialised with a = d = 0.25, b = c = 0. This image  $W_1(T)$  consists of a quarter size copy of T. The image  $W_1(T)$  can then be translated, rotated and sheared by adjusting the coefficients a,b,c and d. The goal is to transform  $W_1(T)$  so that it is lies over part of T. A second sub-copy of the target  $W_2(T)$  is introduced and adjusted until it covers a subset of those pixels in T which are not in  $W_1(T)$ . Overlap between  $W_1(T)$  and  $W_2(T)$  should be as small



as possible for efficiency. A set of contractive affine transformations  $\{W_1, W_2, W_3, \dots, W_n\}$  can be found that characterise the target The image. maps  $\{W_1, W_2, W_3, \dots, W_n\}$ thus determined are stored. The attractor of any IFS code, which uses those maps, will then be visually close to T.



#### 3.3 ITERATED FUNCTION SYSTEM.

Iterated Function System (IFS) Theory can be considered as an extension of geometric transformations [22,28]. IFS consists of a set of contractive affine transformations, denoted by  $\{W_1, W_2, W_3, \ldots, W_n\}$ . For each transformation there is an associated probability, which determines its importance relative to the other transformations, the set of probabilities is represented by  $\{P_1, P_2, P_3, \ldots, P_n\}$ . The number of affine transformations necessary to represent the image depends on the source image. According to the collage theorem, a set of contractive affine transformation can be found that map source images onto a desired target image, also known as the fixed attractor of the IFS transformations.

One of the most well known image used for introducing the IFS is the Black Spleenwort Fern shown in Fig 3.4a. This particular image of the fern can be generated from just four affine transformations. The four affine transformations are shown in equation (3.2) and tabulated in table 3.1a.

$$W_{1}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0 & 0\\0 & 0.16\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\0\end{pmatrix}, \qquad P_{1} = 0.01 \quad \| W_{2}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.2 & -0.26\\0.23 & 0.22\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\1.6\end{pmatrix}, P_{2} = 0.07 \quad \dots (3.2)$$
$$W_{3}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}-0.15 & 0.28\\0.26 & 0.24\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\0.44\end{pmatrix}, P_{3} = 0.07 \quad \| W_{4}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.85 & 0.04\\-0.04 & 0.85\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\1.6\end{pmatrix}, P_{4} = 0.85$$

Transformation  $W_1$  can be simplified into a 1-dimensional equation f(x) = 0, f(y) = 0.16y, this transformation contracts the whole image set into the area occupied by the stem. Affine transformation  $W_2$ , contracts and shifts the whole fern onto the bottom leftmost leaflet whereas transformation  $W_3$  maps the whole fern onto the bottom rightmost leaflet. Matrix A of the last transformation

 $W_4$  can be separated into two parts; a contracting matrix, with a convergence factor of 0.85, and a small rotation to the right. Coefficients b and c of matrix A are responsible for the slight rotation of the fern to the right. Affine transformation,  $W_4$ , maps the area of the stem and the biggest two leaflets onto a contracted space directly above it with a slight shift to the right. Because of the self-similarity of fractals, transformation  $W_4$  maps further contracted copies of the stem and two leaflets just above the previous copy till it reaches the apex of the whole fern, where visually it is just a point. Magnification of this point will reveal a further portion of the fern.

Figure 3.4b is the image of the Sierpinski triangle. Only three affine transformations are necessary to generate this image. The three affine transformations are shown below in equation (3.3) and tabulated in table 3.1b.

$$W_{1}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\0\end{pmatrix}, P_{1} = 0.33 || W_{2}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix}, P_{2} = 0.33 \dots (3.3)$$
$$W_{3}\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.5 & 0\\0 & 0.5\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0.5\\0.5\end{pmatrix}, P_{3} = 0.34$$

The affine transformation  $W_1$  is responsible for generating the pixels in the bottom left hand triangle. Affine transformations  $W_2$  and  $W_3$  affect the bottom right hand and the top triangle respectively. Matrix A, of the three transformations,  $W_1$ ,  $W_2$ ,  $W_3$  represent contractive mappings with a contractive factor of 0.5. Examining the image will reveal the self-similar properties of a fractal set. A portion of the image will be similar to the original fixed attractor, therefore magnification of any one triangle will reveal three smaller Sierpinski triangles similar to the whole image.

The images shown in figure 3.4a and 3.4b could be generated either by the Random Iteration Algorithm or Deterministic Method [28,29]. The following code summarises the random iteration algorithm

set (x	(y, y) = (0, 0);
for N	lo.of.Iterations (
	Choose one of the Transformation according to the probability;
	Apply the Transformation to the point $(x, y)$ to get $(x2, y2)$ ;
	Set $(x, y) = (x^2, y^2);$
	if No. of Iterations $> 10$ , plot (x, y);
}	



W	a	b	с	d	е	f	P
1	0	0	0	0.16	0	0	0.01
2	0.2	-0.26	0.23	0.22	0	1.6	0.07
3	-0.15	0.28	0.26	0.24	0	0.44	0.07
4	0.85	0.04	-0.04	0.85	0	1.6	0.85



W	a	b	с	d	e	f	Р
1	0.5	0	0	0.5	0	0	0.33
2	0.5	0	0	0.5	1	0	0.33
3	0.5	0	0	0.5	0.5	0.5	0.34

#### 3.4 BLOCK BASED FRACTAL CODING (Part I).

The Iterated Function System (IFS) was first applied to digital image compression by A.C. Jacquin and M.F. Barnsley (A.C. Jacquin was a Ph.D student of Dr. M.F. Barnsley). Jacquin outlined a method of applying these functions for block based coding of a digital image in [30,31]. A later paper of J.M. Beaumont [32] from BT Research Lab describes a modified implementation of this technique.

To investigate the performance of such a system, a practical implementation which extracts from both methods is described in the next two sections. Software programs of the block based fractal coding technique were written in Occam and the results displayed on a transputer-based system. The description of the system together with results of the algorithm is presented in Chapter 4.



In image transformations the goal is to find a contractive mapping of an image into itself. The approach mentioned above consisted of trying to match a transformed version of the whole image with a part of itself.

This approach is not very successful as it is often difficult, and in some cases impossible, with a limited class of transformations, to perform the matching. However, it can be observed that it is quite common for some parts of an image to display a similarity with other parts of the image. For example, with a landscape, one part of the sky is likely to look much the same as another part of the sky. With this motivation in mind, the image is divided into a number of smaller sub-images or subpictures and then an attempt is made to match transformed versions (by a process of shrinking, skewing, grey level scaling and rotation) of the sub-images blocks with other parts of the image.



The global transformation of the image into itself is therefore constructed from a combination of local block transformations. Clearly, this is a much easier task than mapping the whole picture onto a part of itself. The process of covering the image with transformed copies of itself is akin to producing a collage of the image. Information available from the coefficients of the transformations can then be used to reconstruct the image (the attractor of the transformations) at another site or location.

#### 3.4.1 Range Blocks.

The input to the system is a digital image and in this work, monochrome images of  $256 \times 256$  pixels were used. The image is divided into smaller blocks called range blocks and larger blocks, known as domain blocks. Square blocks are chosen as it is natural for geometrical transformation from domain block to range block. The size of the range blocks is important, small blocks are easy to analyse and classify geometrically, encoding is accurate and this leads to a robust encoding system (whose performance is steady, even when images are complex with rugged boundaries or fine textures). Larger blocks, however, exploit smoothly varying, or uniform, image areas and lead to higher compression ratios. A range block of  $4 \times 4$  pixels is a reasonable compromise and was used in this investigation.

#### 3.4.2 Domain Blocks.

Suppose a range block of size  $B \times B$  is chosen. The domain pool for this block size can be thought of as all image blocks of size D > B in the image to encode, this is typically very large. There is a need to trim and organise this pool, in order to make the search for an optimal domain block tractable. The domain block pool used was built from  $12 \times 12$  domain blocks, those blocks overlapped such that a new domain block starts every 4 pixels. Each  $12 \times 12$  domain block consists of 9 range blocks and the mean value of the domain block can therefore be evaluated from the means of the range blocks from which it is tiled.

#### 3.4.3 Searching of the Domain Pool.

The available domain pool is very large and the search method must produce a search path which will find a match in the shortest search time. Beaumont[32] shows that samples of matching domain blocks are not far from the original range blocks, their distribution is Gaussian-like. Therefore, self-similarity by part

seems to be a local property. To help in the search, the domain blocks in the pool are classified based on their perceptual geometric features [34]. A simple classification defines two types of blocks; flat blocks and non-flat blocks. A flat block is defined as a block with no significant gradient within the block. The domain blocks pool can be trimmed, flat blocks are useless as domain blocks and can be removed from the pool. This is because a flat block remains a flat block under any of the block transformations.



The search path starts at the domain block nearest to the range block and extends out in a spiral until a satisfactory block match is found. When testing for a block match, the fractal transform can alter 3 qualities of the block data; geometric orientation, grey level offset and the grey level scaling. The difference in pixel values between both blocks is an indication of the goodness of fit. If the error becomes greater than a given threshold even after searching through the eight isometric transformations for the domain block, the test is terminated as a failed match and the search move to the next domain block. If the whole codebook is searched and a match is not found, then the error threshold is allowed to increase and the codebook is searched again. When coding at about 1 bit/pixel, the error threshold very rarely has to be increased.

Therefore, a whole set of transformations can be found to describe the whole picture in terms of itself. At the decoder, given a random picture as a starting point, the original picture can be recovered by iteratively executing the above set of fractal transforms.

#### 3.4.4 Fast Decoding.

In Jacquin's method, the decoder starts with a blank image with only the mean values of all the flat range blocks. It takes 6-8 iterations before the image converges towards an acceptable reproduction of the original image.

If the starting image is a picture close to the original, then the decoder will converge quickly. If range block means are transmitted for every type of block, then the decoder has enough information to compose a picture remarkably close to the original. Consider a picture size  $256 \times 256$ . It will require  $64 \times 64$  range block means, i.e. a smaller version of the original picture. At the decoder, this picture is enlarged, using interpolation, back to the original size  $256 \times 256$  pixels. This looks like a blurred version of the original, and is used as the starting point for the fractal iterations. In practice, after one or two iterations, the decoded picture will converge.

#### 3.5 BLOCK BASED FRACTAL CODING (Part II).

One sub-section will describe in detail possible domain block transformations for matching domain blocks to range blocks. Another sub-section will concentrate on how range blocks can be encoded. Non-flat range blocks can be characterised by scaled down versions of matching domain blocks, whereas each flat range block can simply be represented by a pixel value.

Comparison between range and contracted domain blocks can either be performed in the pixel domain or transform domain. The transform domain was preferred as high frequency details which are visually insignificant can easily be removed. Switching among the eight isometric transformations is easy in the transform domain, which is an advantage. The last sub-section explains and tabulates the number of bits required to represent each fractal transform.

#### **3.5.1** Domain Block Transformations.

Before the domain blocks can be compared with the range block, it needs to undergo a transformation. The general form of the transformation for a block can be separated into two basic parts, a dilation and an isometric transformation. Each range block can therefore be described by a dilation, which is simply a translation followed by a spatial contraction, from a domain block to the range block and a geometric transformation which exhibits isometry within the range block.

#### Form of the Dilation.

Range blocks are of size  $B \times B$ , and domain block of size  $D \times D$ , with D > B. Domain blocks must be larger than range blocks to satisfy the contraction mapping criteria of the Iterated Function System. For this implementation, domain blocks are restricted to squares and can be located anywhere in the image. Figure 3.7 shows a domain block and the contracted domain block. The nine shaded pixels were used to generate the shaded pixel on the contracted block. Each domain block is three  $\times$  three times the size of a range



block, D = 3B, therefore, the pixel values of the contracted image on the range block are simply given by;

$$r_{i,j} = \frac{(d_{4i,4j} + d_{4i+1,4j} + d_{4i+2,4j} + d_{4i,4j+1} + d_{4i,4j+2} + d_{4i+1,4j+1} + d_{4i+1,4j+2} + d_{4i+2,4j+1} + d_{4i+2,4j+2})}{9} \dots (3.4)$$
where  $i, j \in \{0, 1, 2, 3\}$ 

This is computationally attractive because it consists of averaging with equal weights and this can be calculated using just integer mathematics.

The dilation used to map the domain block to the range block, is therefore the composition of a uniform contraction by one-third ( $s = \frac{1}{3}$ , with an invariant point at one corner of the domain block), and a translation that takes the bottom left corner of the domain block to the bottom left corner of the range block.

#### Form of the Isometric Transformation.

There are eight elementary isometrics that are directly inspired from the isometrics of a square. Isometry simply shuffles pixels within the range block, in a deterministic way. It is an equivalence relationship and it preserves topological properties. The eight possible isometric transformations to be used in this work are shown in figure 3.8 and described in table 3.2.



Cases	Possible Canonical Isometrics Transformation.
<i>i</i> <sub>1</sub> -	Identity (no change).
i <sub>2</sub> -	Rotation around the centre of block, through -90 degrees. (+270 deg.)
i, -	Rotation around the centre of block, through +180 degrees.
i, -	Rotation around the centre of block, through +90 degrees.
i, -	Orthogonal reflection about mid-horizontal axis of the block.
i <sub>6</sub> -	Orthogonal reflection about the off-diagonal of the block.
i <sub>7</sub> -	Orthogonal reflection about mid-vertical axis of the block.
<i>i</i> <sub>s</sub> -	Orthogonal reflection about the main diagonal (i=j) of the block.
	Table 3.2 Isometric Transformation for a square block.

#### 3.5.2 Testing for a Block Match.

Range blocks are classified according to their perceptual geometric features, either as a *flat* or *non-flat* block.

#### Range block is a flat block.

A *flat block* is smooth, with no significant gradient within the block. No search is necessary and  $B_i$  can simply be approximated by a uniform block. A single pixel value equal to the average pixel intensity of the range block is sufficient to represent a flat block.

#### Range block is a non-flat block.

A non-flat block can be classified further into two main types, an edge block and a midrange block. An edge block presents a strong change of intensity across a boundary which runs through the block. A midrange block has a moderate gradient but no definite edge. It has no well-defined orientation and finely textured blocks belong to this category.

Every element in the domain pool of the same type as the range block is scanned. The domain block in the search path can be compared with the current range block by transforming the domain block;

$$\mathbf{R}_{i} = i_{n} \left[ g_{i} \hat{\mathbf{D}}_{i+j} + o_{i} \right] \qquad \dots (3.5)$$

where  $\mathbf{R}_i$  and  $\hat{\mathbf{D}}_{i+j}$  are matrices of the elements in the range and contracted domain block respectively.  $g_i$  is a scaling factor smaller than or equal to one, this value is allowed to take on values in the set  $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$  and is chosen so that the range block and matching domain block have the same dynamic ranges.  $o_i$  is computed so that the average grey level of the range block and the scaled down domain block are the same.

$$o_i = \overline{R}_i - g_i \hat{D}_{i+j} \qquad \dots (3.6)$$

where  $\overline{R}_i$  and  $\hat{D}_{i+j}$  are the mean values of the range and matching domain blocks respectively.

The last variable in the transformations is isometric,  $i_n$ , where *n* is from 1 to 8 and represents one of the eight isometrics which minimises the distortion measure. For an edge block with no well-defined orientation, the original isometric,  $i_1$ , is a good enough representation and there is no further need to search the other forms of isometrics.

#### 3.5.3 Block Matching in the Transform Domain.

The range block and the domain blocks in the transformation pool can be examined by comparing the pixels between both blocks. From the study of the human visual system, it is known that the human eye is more sensitive to coding errors at some frequencies than others and this can be exploited if data is moved into the transform domain.

#### Hadamard Transform.

The Hadamard transform is based upon the Hadamard matrix , which is a square array of plus and minus ones and whose rows and columns are orthogonal . An ordered, normalised  $4 \times 4$  Hadamard matrix is shown here,

Although it is a sub-optimum transform, it is important because it is computationally attractive. When searching for a matching domain block, the AC coefficients of the two blocks are compared in the order of their distance from the DC coefficient (in the standard zigzag scan order). If the summed error exceeds a predefined threshold, the domain block goes through to the next isometric transformation (rotation or reflection) and the AC coefficients are compared again.

As shown in Figure 3.9, there is no requirement to perform the Hadamard transformation again. When switching from one Case to the next, we can just take the transformed matrix and transpose it, and then negate the coefficients in the even rows of the matrix.

Fig 3.9: Typical set of results for the eight Isometric Transformation.

$$W_{n} = \begin{cases} (W_{n-1} \cdot A)^{T}, & n = \{2, 3, 4, 6, 7, 8\} \\ (W_{n-1})^{T}, & n = 5 \end{cases} \qquad \dots (3.8)$$
  
where  $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$ 

i

This would reduce the time required in searching for a matching domain block. The above condition holds for all cases except when switching from Case 4 to Case 5, when it is only necessary to transform the matrix.

#### Quantisation Error Threshold.

As in Beaumont's method, the varying sensitivity of the eye to frequency information is exploited. A different error threshold is used for each coefficient when testing for a block match. A table of the relative threshold error matrix for a luminance signal is shown below;

Quantisation Thresholds, 
$$G = \begin{pmatrix} 1 & 4 & 9 & 17 \\ 4 & 6 & 15 & 22 \\ 9 & 15 & 25 & 29 \\ 17 & 22 & 29 & 29 \end{pmatrix}$$
 ... (3.9)

These thresholds are changed depending on the quality of the picture required but the relative values remain unchanged. The thresholds allow the use of more compact, and hence less accurate, coding in the frequency bands to which the eye is less sensitive. The above matrix has been derived from similar values used for DCT coefficients in the ISO JPEG still picture standard [2].

#### 3.5.4 Bits required for each Fractal Transform.

#### Mean Value and Type of Range Blocks.

Every range block in the image has been organised into two main types. Therefore, one bit is sufficient for classification of range blocks into flat blocks and non-flat blocks.

Each pixel in a monochrome image is represented by 256 discrete levels. The mean value which is a weighted average will also be represented by 256 discrete levels or 8 bits. Therefore, one byte is required to transmit the mean value of every  $4 \times 4$  block.

If the difference between blocks is transmitted instead, a further reduction can be achieved. This takes into account only the relationship between adjacent blocks in the same line. The mean value of the next block is subtracted from the previous block and only the differences need to be transmitted. To prevent a delay during the initial startup of every line, a mid value of 128 (grey level) is used at the beginning, the first block then subtracts the 128 from its mean value and transmits the difference. Six bits will then be sufficient to transmit the difference in adjacent blocks.

#### Position of the Matching Domain Blocks.

For every non-flat range block, a matching domain block from the same image will be found. The horizontal and vertical offset, which identify the position of the matching domain block can be represented by just 6 bits for each direction. This is because there are  $256 \times 256$  pixels which make up  $64 \times 64$  contracted blocks. The blocks are non-overlapping and there are only 64 available block positions in either direction.

#### Scaling Factor, Offset of Range and Matching Domain Blocks.

As described in an earlier section,  $o_i = R_i - g_i \hat{D}_{i+j}$ , where  $o_i$  and  $g_i$  are the grey level offset and scaling factor respectively. In the original method, both the scaling factor and the grey level offset were transmitted as parts of the fractal transform. As can be seen from the above, the grey level offset can be computed at the decoder from the scaling parameter and the associated domain block and range block means. The domain block mean can be evaluated from the means of the range blocks used to tile it. Therefore the increase in the bit rate due to every range block mean being transmitted is offset by not having to transmit the grey level offset.

The scaling or gain factor has been constrained to take on values in the set (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8). Therefore, three bits are sufficient for representing the scaling or gain factor.

#### Affine Transformation Matrix.

The 2×2 matrix A of the affine transformation used in this work is in a standard format. Since the transformation is from a  $12 \times 12$  domain block to a  $4 \times 4$  block, the contraction factor is always  $\frac{1}{3}$ . The matrices used for the eight isometry transformations are always the same and they are shown in equation (3.10);

$$Case \ 1:- \ A_{1} = \begin{pmatrix} \frac{1}{3} & 0\\ 0 & \frac{1}{3} \end{pmatrix} \qquad Case \ 2:- \ A_{2} = \begin{pmatrix} 0 & \frac{1}{3}\\ -\frac{1}{3} & 0 \end{pmatrix} \qquad \dots (3.10)$$

$$Case \ 3:- \ A_{3} = \begin{pmatrix} -\frac{1}{3} & 0\\ 0 & \frac{1}{3} \end{pmatrix} \qquad Case \ 4:- \ A_{4} = \begin{pmatrix} 0 & -\frac{1}{3}\\ \frac{1}{3} & 0 \end{pmatrix}$$

$$Case \ 5:- \ A_{5} = \begin{pmatrix} \frac{1}{3} & 0\\ 0 & -\frac{1}{3} \end{pmatrix} \qquad Case \ 6:- \ A_{6} = \begin{pmatrix} 0 & \frac{1}{3}\\ \frac{1}{3} & 0 \end{pmatrix}$$

$$Case \ 7:- \ A_{7} = \begin{pmatrix} -\frac{1}{3} & 0\\ 0 & \frac{1}{3} \end{pmatrix} \qquad Case \ 8:- \ A_{8} = \begin{pmatrix} 0 & -\frac{1}{3}\\ -\frac{1}{3} & 0 \end{pmatrix}$$

As can be seem from the above equations, the affine transformation matrix A need not be sent. A 3-bit representation of the matrix is sufficient and the receiver can construct the matrix once it knows which isometric transformation is needed to match the range block. (Only 8 possible cases were considered, and 3 bits will be sufficient for representing the isometric transformation of a square).

Block Type	Parameters		Bits	Total No. of Bits
Flat Blocks	Block Type :-	$T_i$	1	
	Mean Value :-	$R_{i}$	6	7 Bits
	Table 3.3a: Bits R	equired for	Flat Range Blo	cks

Block Type	Parameters	Bits	Total No. of Bits		
Non-Flat	Block Type :- $T_i$	1			
Blocks	Mean Value :- $R_i$	6			
	Position : - x axis	6	25 Bits		
	Position :- y axis	6			
	Scaling Factor :- g	3			
	Isometric :- in	3			
Table 3.3b: Bits Required for Non-flat Range Blocks					

The algorithm as described in the last two sections is implemented as a transputer process and a description of the simulation and results are presented in the next chapter.

# CHAPTER 4



## CHAPTER 4 : IMPLEMENTATION AND RESULTS OF AN IFS SYSTEM.

This chapter will define the hardware system used for testing the block-based fractal coding algorithm explained in Chapter 3. Occam programs written for encoding and decoding of images will be described with the aid of flowcharts. Some images resulting from this implementation are shown in section 4.3 together with some comments. The last section will conclude with a general discussion on the method of fractal coding.

#### 4.1 TRANSPUTER-BASED DISPLAY SYSTEM.

The display system can be grouped basically into 3 main parts:-

- (i) Transputer-based Board (Controller).
- (ii) Video Frame Store.
- (iii) Digital-to-Analogue Board.





#### 4.1.1 Transputer Based Board (Controller).

The Transputer with its 2 Mbytes of external memory and associated components are on a single board. This board was developed in-house and an area of memory is allocated for external uses with physical sockets on the board for possible EPROMs. This is useful after the program development phase, for implementation of the fractal coding algorithm as an embedded system without the host operating system.

Input for the display system can be feed in through the transputer links which are accessible to the outside world. A software program running on the transputer can be used to access and if necessary manipulate the data available. This is a general purpose transputer board and during development stages the encoding program uses this board for executing the algorithm.

#### 4.1.2 Video Frame Store.

The main components on the frame store consist of the video system controller and video memory (VRAM). This frame store has a memory area of 0.75 Mbytes  $(512 \times 512 \text{ pixels} \times 24 \text{ bits depth})$ . The video memory which makes up the frame store is basically organised as another part of the transputer memory map. Thus, the transputer can access the video memory simply by writing to the specific memory location. VideoRAM are dual ported RAM, thus it is possible to read from and write to the memory at two different ports simultaneously. At one port, the transputer can write directly to the RAM, while at the same time, the data in the frame store can be read out continuously to the DAC board in real time generating 25 frames per second. The video system controller (VSC) on the display unit generates all the timing and control requirements for the RAM. It is also responsible for generating the video synchronisation and blanking signals necessary to control a monitor.

#### 4.1.3 Digital to Analogue Board.

The Digital to Analogue Converter (DAC) board embodies a triple 8-bit Digital to Analogue converter and analogue amplifiers for the video signal. Digital video data are converted into its analogue equivalent and fed directly into a monitor. The analogue signals could either be in YUV or RGB format but YUV signals need to be converted before feeding into the RGB monitor.



#### 4.1.4 Utilisation of the Display System.

Block based fractal coding is an asymmetrical process, the encoding of an image takes up much more time as compared to the decoding. If necessary, more processing power can be allocated to the encoder as the algorithm itself is suitable for parallel processing. Searching the domain blocks pool for a match to each non-flat range block is independent and can therefore be done in parallel but at an increase in cost.

The transputer-based display system is used for both the encoding and decoding algorithm. In program development stages, the encoding operation uses the transputer available on the board as a processor. Digital images are made available to the host operating system as raw bit files (similar to raw .ppm files but without the file header). The transputer will access the image data and processing of the information will generate a set of fractal transform codes which are stored in the output file. During encoding, the original image can be displayed on the monitor, if required.

For decoding, the transputer board reads the encoded file and extracts the transmitted fractal transform codes and mean value of every range block. An initial image can be generated by using the mean value to represent all the pixels in a range block and this is repeated for every mean value. The image is stored in a part of the memory map which maps onto the VideoRAM. Video information can thus be displayed directly on a monitor. The extracted fractal transforms, consisting of coefficients of the affine transformations are then used to improve the image. The image will progressively get better as the number of iterations increases.

#### 4.2 PROGRAM FLOWCHARTS.

Most transputers will probably end up in embedded computer systems but during the program development process, some operating system facilities must be available, such as access to a file on a disk, terminals, text editors, high level language compilers and debuggers. The simplest way is to use an existing machine as a host, running a server program that communicates with the Transputer system.

The INMOS Transputer Development System (TDS) is one such system. The DOS version has a server that runs on a PC Host, communicates through a transputer link with a plug-in board interfaced by a link adapter to the PC bus.



The principle language provided with the TDS was Occam, a high level language that was designed for, and with, the transputer. Occam provides most of the features expected in a high-level language but programmers experienced in Pascal or C will find Occam unusual; there is no recursion, no structures or records except in input and output, no dynamic memory allocation, no userdefined types. On the other hand, Occam provides access to some of the transputer's facilities in a very clear and simple fashion.

#### 4.2.1 Encoding of Monochrome Images.

The program for the encoding of  $256 \times 256$  monochrome images was written and a flowchart is shown here in Figure 4.2. What follows is a brief description of the program, which can be divided into three main parts.

#### Preparing the Domain Pool.

The variables used are initialised and a  $256 \times 256$  uncompressed image is read from the disk. This digital image is then broken up into  $12 \times 12$  blocks. A new 12  $\times$  12 domain block is defined for every separation of 4 pixels in either direction. Therefore, two adjacent blocks will have two-thirds of their pixels overlapping. Each domain block is reduced from  $12 \times 12$  to  $4 \times 4$  pixels by a process of equalweight averaging. Domain blocks are then converted into the transform domain by the Hadamard matrix. For ease of comparison, the domain blocks in the pool can be normalised by dividing the coefficients in each block by its standard deviation. The final process for each block is a quantisation stage where the AC coefficients further away are quantised with a larger threshold. The quantised values are then stored into another 2-dimensional array which will be used when searching the domain pool.

#### Preparing the Range Blocks.

The original image is divided into  $4 \times 4$  blocks and the same process of Hadamard transformation, standard deviation and quantisation are applied to each  $4 \times 4$  range block.





Each range block is checked to see if it is a flat block (AC coefficients are zero). If it is a flat block, only the mean value of the block needs to be transmitted or stored. If not, a spiral search which starts at the nearest domain block is conducted in order to match domain blocks in the search pool to the range block. If a match is found, the position of the matching area is recorded, together with the scaling factor, mean value of the block and the isometric transformation number.

#### 4.2.2 Decoding of Monochrome Images.

A flowchart of the decoder is shown in figure 4.3 and a brief description follows;

#### Prepare the Startup Image.

The variables to be used are initialised. Compressed video information will be extracted from the output file generated by the encoder. Mean values transmitted



for every block are used to construct a startup image, each mean value is used to represent all the pixels in a  $4 \times 4$  pixel block. This initial image is relatively close to the actual image, it looks like a blurred version of the original.

#### Apply Transformations.

For flat blocks in the image, only the mean values were transmitted. Modification to the range blocks is not necessary and no affine transformations were transmitted. For every non-flat block, extra parameters can be extracted from the file. The x-y offset, scaling factor and coefficients of matrix A will form the required 2-dimensional affine transformation. This transformation will be used to extract information from the matching domain block so as to modify the pixels in the range block. One or two iterations of all possible affine transformations in the image will be sufficient to reconstruct an image that is visually close to the original image.

#### 4.3 RESULTS FROM THE SIMULATION.

#### 4.3.1 Test Results.

Three test images were selected for evaluating the block based fractal coding algorithm. Each monochrome image has been captured at a resolution of  $256 \times 256$  pixels, with 256 grey levels representing each pixel.

- Image A "Beauty" :- Head and shoulders image of a former Miss America.
- Image B "Fruits" :- A close up of an apple and an orange.
- Image C "Earth" :- Satellite image of the earth.

For each test image, the following items are provided, except for the first set of images "*Beauty*" where an additional image, the startup image, is shown. This image is constructed from the mean value of each block.;

- The original test image.
- The decoded image, generated from a fractal approximation of the image.
- A magnified portion of the decoded image.
- The error image, showing the differences between the original and decoded images. To be visually noticeable, the differences are amplified four times.
- A result sheet is provided for each set of images.



#### 4.3.2 Understanding the Result Sheet.

For each test image, a result sheet is provided which contains the following information;

- "Original Image" :- This lists the basic characteristics of the digital image under test. It includes name, resolution and the number of grey levels to which it is quantised.
- (ii) "Encoding Specifications" :- This consists of details such as the size of range and domain blocks, domain block shift and the type of classification for blocks.
- (iii) "System Performance" :- This tabulates the global proportions of flat and non-flat (edge and midrange) blocks in the original image which provide an indication of the complexity within the image. File size before and after compression are compared. Lastly subjective remarks are made about the fidelity of the decoded image compared with the original.

#### 4.3.3 Analysis of the Results.

The performance of the block based IFS encoding and decoding system used to carry out the simulations can now be reviewed. This is a lossy compression technique and the reconstructed image will therefore be an approximation of the original. As can be seen from the decoded images presented, the fidelity of the decoded images compared with the respective originals are good. Textures in the images are generally well preserved, except when a matching domain block cannot be found and the error threshold has to be increased. Some blockiness is noticeable which arises mostly from the encoding of flat blocks as uniformly grey blocks. It disappears almost completely for blocks which are encoded as non-flat blocks.

#### Image A: "Beauty".

Figures 4.4(a) - 4.4(d) are images of "*Beauty*". Figure 4.4(a) shows the original  $256 \times 256$  image with 8 bits per pixel to represent intensity. Figure 4.4(b) shows the same image after compression and decompression. This figure consists of the decoded image and a magnification of the area around the eyes. The decoded image is visually close to the original image. Textures are, in general, well preserved. Sharp, contrasting contours, whether they are smooth (outline of face, nose), or rugged (outline of hair), are very accurately preserved. The only areas where the performance is degraded occur when a range block cannot find a matching domain block within the image and the error threshold has to be



increased. A good example of this is in the left eye of the reconstructed image where the error is visually noticeable. The right eye however, of the reconstructed image has quite good reproduction and is visually very close to the original. Figure 4.4(c) is the error image, this shows the variations of the difference (amplified four times) between the original and reconstructed images. The errors are mainly scattered within the areas where matching domain blocks are difficult to find. It is near areas where there is a great deal of detail, like the area around the eyes and the mouth. Figure 4.4(d) shows the startup image used, this looks like a blurred version of the original. The startup image is assembled from the mean values of the range blocks and with this image the number of iterations required for convergences is reduced. A typical image will be visually close to the original after one or two iterations.

#### Image B: "Fruits".

Figures 4.5(a) - 4.5(c) are images of "*Fruits*". Figure 4.5(a) shows the original  $256 \times 256$  image with 8 bits for each pixel to represent intensity. Figure 4.5(b) shows the same image after compression and decompression. Textures are quite well reproduced. The sharp edges are not very well preserved and some blockiness is visible in the image. This is due to the limited domain pool available. From the results sheet, it can be seen that 60% of the image is made up of flat blocks and 40% as non-flat blocks. A higher percentage of the domain blocks are therefore likely to be flat. Non-flat range blocks will then have difficulty in finding a match in the reduced domain pool. But with more flat blocks, the compression and decompression time will be reduced.

Figure 4.5(c) shows the error image. With a reduced domain pool, matching domain blocks are difficult to find. Errors are scattered around the circumference of the apple whereas errors within the orange appear in the texture. Errors in texture are not visually as prominent as edge errors.

#### Image C: "Earth".

Figures 4.6(a) - 4.6(c) are satellite images of the "*Earth*". Figure 4.6(a) is the original digital image of resolution  $256 \times 256$  pixels, with 8 bits per pixel to represent intensity. Figure 4.6(b) shows the same image after compression and decompression. The image consists of the decoded image and a magnification of an area within the image. Some blockiness is noticeable. The sharp contours within the image, whether they are smooth or rugged, are reasonably well preserved. The block statistics show that 59% of the blocks are flat blocks,

therefore blocks available to form the domain pool are limited, this reduces the choice of domain blocks but it also reduces the time required for compression. Figure 4.6(c) is the error image. Errors are well scattered within the image not concentrated within any particular area.

#### 4.4 SUMMARY AND CONCLUSION.

Chapter 3 commenced with a brief description of fractal and some associated theory, followed by an introduction to the Iterated Function System and ended with a description of a digital image coding system based on block based fractal coding. Chapter 4 has described the transputer-based display system used in evaluating the algorithm together with a brief description of the Occam program. Compressed images resulting from this algorithm are shown in this chapter together with comments.

All practical implementations of fractal coding reported in the professional literature, including the one by A.C.Jacquin and M.F.Barnsley, work by dividing the original image into sub-blocks and then performing a mapping operation on each of those sub-blocks. This method is explained in chapter 3 together with one possible implementation of this block-based fractal coding algorithm. By dividing the image into small blocks, coding of natural images is possible but the number of affine transformations increases proportionally with the number of sub-blocks in the image. As the number of affine transformations increases, the amount of information to be transmitted will also increase. Compression, therefore, will be greatly reduced. Typical compression ratios quoted for this method are comparable with standard algorithms, e.g. JPEG, but nowhere near the enormous compression ratios quoted in the popular technical press. Possible further extensions to this algorithm include;

- i) The geometry of the sub-blocks does not have to be square, other geometries, which might yield less blockiness, could be considered.
- ii) The notion of self-transformability should be further investigated. It is not clear how localised a property this is for a given class of images.
- iii) A move from block based fractal coding towards feature based fractal coding techniques should achieve greater compression ratios. This will create another class of problems, i.e. what is the definition of a feature and how the threshold for segmentation should be drawn.



Hardware implementation of the block-based fractal coding needs to take advantage of the asymmetrical nature of this algorithm. Encoding is computationally intensive and parallel implementation can be exploited. Searching for the affine transformation of each sub-block is independent of the others and this part of the algorithm can be processed in parallel. Transputers are an ideal choice for this and a network of transputers can be used to encode the images. Decoding and displaying the results can be done using the transputer based display system.

Results of the simulation are visually acceptable. Textures and edges in the images are generally well preserved but some blockiness is visible. When coding at 1 bit/pixel, a matching domain block for the range block can easily be found within the image without increasing the error threshold. An original image with a high percentage of flat blocks will have a reduced domain pool. This will make the search for a matching domain block more difficult. Edges in images B and C are not as well reproduced because of this reduction in the size of the domain pool. Compression ratios of 6-9:1 are achievable for monochrome images. This compression ratio is comparable with standard algorithms but is not as high as was initially expected. Improvements may be achieved if the suggested changes could be implemented.





ORIGINAL IMAGE	C:
Name :	Beauty
Resolution :	256 × 256 pixels
Grey Level :	256 levels (0-255)
Encoding Specificat	ion:
Partition	
Range Blocks	4×4 Blocks
Domain Pool	
Domain Blocks:	$12 \times 12$ Blocks
Domain Block Shift:	4 pixels in either direction
Classifications:	Flat and Non-Flat Blocks
System Performance	2:
Block Statistics:	16.8% Flat Blocks
	83.2% Non-Flat Blocks
File Size (Before):	65536 Bytes (64 Kbytes)
File Size (After):	10912 Bytes
General Remarks	Good reproduction of sharp edges (outline of hair, eyelids,
	face and background). Destruction of some fine texture and
	detail (around the left eye, a good match cannot be found).
	Table 4.1 Result Sheet for Image 1: "Beauty"

Ň

1




÷.

ORIGINAL IMAGE:				
Name :	Fruits			
Resolution :	256 × 256 pixels			
Grey Level :	256 levels (0-255)			
Encoding Specification:				
Partition				
Range Blocks	4 × 4 Blocks			
Domain Pool				
Domain Blocks:	$12 \times 12$ Blocks			
Domain Block Shift:	4 pixels in either direction			
Classifications:	Flat and Non-Flat Blocks			
:				
System Performance:				
Block Statistics:	60.8% Flat Blocks			
	39.2% Non-Flat Blocks			
File Size (Before):	65536 Bytes (64 Kbytes)			
File Size (After):	7308 Bytes			
General Remarks	Acceptable reproduction of sharp edges. Some blockiness			
	visible in the tip of the apple.			
Table 4.2 Result Sheet for Image 2: "Fruits"				





ORIGINAL IMAGE:				
Name :	Earth			
Resolution :	256 × 256 pixels			
Grey Level :	256 levels (0-255)			
Encoding Specification:				
Partition				
Range Blocks	4 × 4 Blocks			
Domain Pool				
Domain Blocks:	$12 \times 12$ Blocks			
Domain Block Shift:	4 pixels in either direction			
Classifications:	Flat and Non-Flat Blocks			
System Performance:				
Block Statistics:	59.3% Flat Blocks			
	40.7% Non-Flat Blocks			
File Size (Before):	65536 Bytes (64 Kbytes)			
File Size (After):	7430 Bytes			
General Remarks	Some blockiness noticeable.			
Table 4.3 Result Sheet for Image 3: "Earth"				

# CHAPTER 5 ▷▷▷

## CHAPTER 5 : TRANSMISSION OF PACKET VIDEO OVER AN ATM NETWORK.

#### 5.1 ASYNCHRONOUS TRANSFER MODE.

Asynchronous Transfer Mode (ATM) has been widely accepted as the transfer mode for further Broadband-ISDN [35]. It is a form of fast cell switching similar in concept to protocols used over existing data networks, such as X.25 and frame relay but with fixed short sized frames or cells. ATM can carry virtually all current types of traffic, including telephony, facsimile, X.25 packet, current (Narrowband) ISDN, kilostream, etc. ATM cannot claim to be a superior medium for any of these, but it provides a good compromise suitable for the multiplexing of many diverse services together. It is especially good for;

- Constant-rate services which do not conform to the standard rates provided by the synchronous network.
- Bursty services, where the data rate required varies over time.

Some believe that ATM should be used for the whole Broadband-ISDN network, eventually replacing all other types of switching fabric. Others hold that ATM should become part of a mixed strategy, which will also employ Synchronous Transfer Mode, e.g. Synchronous Digital Hierarchy (SDH), and some Narrowband switching for basic telephony.

ATM traffic is carried in 53 octet (byte) packets called cells. The ATM cells are of fixed length with five octets being consumed by header information. This leaves 48 octets free for carrying data.

٢



#### ATM Cell Header.

The ATM cell header provides functions including routing (virtual path and virtual channel), flow control, payload types, cell loss priority, header errors control and empty cell indication. The header is formatted in one of two ways. The first applies at the User Network Interface (UNI) between the customer and the exchange, while the second applies at the Network Node Interface (NNI) between nodes of the ATM network.



#### Cell Loss Priority.

What is of interest here is the Cell Loss Priority (CLP) bit in the ATM cell header. The CLP bit can be used to tag cells that have violated their permitted data rate. The cell will still proceed through the network but with an increased risk of loss. Cells that exceed the negotiated limit will be discarded when necessary and a heavy cost penalty could be imposed for transmitting tagged cells.

Another possible use of the CLP bit is to indicate the priority of the cell being transmitted. Cells with lower priority can be discarded in preference to other cells when there is congestion. This could be used in the transmission of video information to separate the bulk of the picture information from essential framing

information. This bit could also be used for two layer video coding algorithms to separate the high priority base layer from the lower priority enhancement layer. The use of the CLP bit as described in the second scenario would be preferred by groups with an interest in video transmission and it will be interesting to see what the standardisation body decides.

#### 5.2 CELL STREAM CONTROL OF PACKET VIDEO.

Protective measures should be implemented to minimise the loss of cells in the network during the overload period by controlling the cell streams at the transmitter end. ATM capabilities are affected by delay jitters and packet lost probability and these are important considerations for video applications. Cells may be lost due to random bit errors in a cell header or to network control when traffic is congested. Cell loss due to network congestion is considered the more serious problem to overcome.

For single layer video codecs where all cells generated by the coding algorithm are of the same priority, it is not possible to implement a control scheme to selectively drop cells. But to reduce the overload periods, a control scheme can be implemented to discard all arrivals once they exceed the maximum delay.

It would be advantageous if the output of the video codec could be separated into high priority data and low priority data, where the loss of low priority data might only cause a modest picture degradation. With such hierarchical coding, a congestion control scheme could be employed that selectively discards low priority data to alleviate the possibility of overload. This would significantly prolong the underload period and reduce the number of high priority losses in the overload period. To further reduce high priority arrivals in overload period, a control scheme to block all arrivals which exceed the maximum delay could also be implemented.

#### 5.3 TREATMENT OF VIDEO CELL LOSS.

At the receiver, cells are checked for their validity. Lost cells are detected at the segmentation and reassembly layer (SAR) and recovery of the video cells can start. Many compensating measures have been proposed recently to make cell loss subjectively less perceptible. Cell loss recovery mechanisms can be categorised into two groups:



• *Error Concealment:* A technique where a missing cell is detected and replaced by an approximation to its content derived from existing information in the data stream. This requires no extra coding information.

#### 5.4 ERROR CORRECTION.

#### 5.4.1 Selective Retransmission.

If the receiver has a return path available to send messages to the source, it is then possible for the receiver to call for the retransmission of cells that contain errors. This scheme is commonly known as retransmission error control. It is a viable option for non-time critical operation like file transfer over a computer network but traditionally, error control by retransmission has not been favoured for time critical, interactive services such as voice and video because of the time delay between detection of an error and the eventual arrival of the retransmitted data.

If the loss is due to network congestion, a likely scenario for an ATM network, asking for retransmission would add to the congestion and therefore would not be recommended.

In a wireless network, when a mobile station moves outside the base station's coverage area, it is necessary to handover the call to another base station. Errors occur during the handover period and they could be corrected by retransmission when the radio link has been re-established. This could be feasible for MPEG type video where important information in the I-frame (DCT coded frame) could be retransmitted and used to help in the decoding of subsequent P and B-frames.

#### 5.4.2 Forward Error Correction.

Many important block codes for random error correction fall into the family of BCH codes, named after the researchers Bose, Chaudhuri and Hocquenghem. A straight forward implementation of a BCH code over consecutive cells will require a large number of overhead bits and rather complex implementation. A simpler implementation is shown in figure 5.3 [36]. The data are stored in a 2-dimensional array with each row consisting of 48 bytes. Each column of data is covered by a simple BCH code and the error correcting bits are appended to the



end of that column. After calculating the BCH codes for every column, each row is packaged into an ATM cell and transmitted. Therefore, the information bits and error correction bits are transmitted in different cells.

The CCITT Rec I.363 - "B\_ISDN ATM Adaptation Layer specification", will contain an optional octet based (128, 124)

User Data		Transmitted Cells
for Correcting Bits	M	
	User Data for Correcting Bits - 48 Bytes	User Data for Correcting Bits - 48 Bytes

Reed-Solomon code, with an interleaving matrix of  $128 \times 47$  octets, resulting in a delay of 128 cells. This code will be able to detect up to four cell errors in a row of 128 cells. In the MPEG-2 committee draft, there is also an option of implementing Reed-Solomon codes to protect the data stream.

#### 5.5 ERROR CONCEALMENT.

#### 5.5.1 Bit/Byte and Block Interleaving.

To protect against the loss of a continuous stream of data, bit/bytes within a cell are taken from various parts of the data stream based on the interleaving algorithm. To protect against burst errors, data in neighbouring cells should be separated as far as possible, but to buffer up a chunk of data before segmentation would increase the delay and a compromise must be struck.





Implementation of block interleaving works above the segmentation and reassembly layer and is related to the operation of the picture coding algorithm. Usually, an image is divided into smaller segments and each consecutive coded segment is sent to the SAR layer. However, as shown in figure 5.4 when shifting to the next block in the horizontal direction, it alternates between the two slices. Each cell is filled with an integer number of blocks and the remaining byte positions in the cell are stuffed with a fixed pattern. Block interleaving isolates the transmission errors and error concealment algorithms can be applied at the receiver to correct for the errors.

#### 5.5.2 Intra-Frame Reconstruction.

It is well known that the spectra of common images have large areas with low frequency characteristics. In the spatial domain, this is reflected by the abundance of flat areas. A course in picture composition or photography would reveal the reason for such flat areas. Most images have a prominent subject which will convey a message to the viewer. Areas around the subject should draw attention or at least not distract the viewer from the main attraction, therefore these background areas are usually out of focus, of a constant hue and contrast or a constant structure.

Natural images therefore tend to exhibit a high correlation between neighbouring pixels and thus it is possible to implement error concealment techniques for recovery of areas lost due to transmission errors.

Techniques of error concealment in the pixel domain include replacing the lost areas with linearly interpolated values of the surrounding areas within the same frame. Methods investigated for DCT-based images include one [37] which assumes the loss of only a partial set of the transform coefficients and reconstruction is then possible with information from the correctly received transform coefficients in the same block and from information of the pixels surrounding the error block. Another method [38] assumes the data are from an autoregressive processes and using a predictive signal model, exploits the information contained in the correct pixels neighbouring the pixels to be restored.

#### 5.5.3 Inter-Frame Reconstruction.

Depending on the coding algorithms used, different error concealment algorithms can be used. Predictive frames in MPEG and H.261 video coding algorithms consist of a set of motion vectors which are estimated by comparing two consecutive frames. Motion vectors that are lost can be estimated from the neighbouring motion vectors, a majority vote of the neighbouring motion vectors will give a good reproduction of the direction and amplitude of the lost motion vector. Errors in intra-coded frames can be corrected using the methods described in the previous section.

#### 5.6 ERROR CONCEALMENT OF JPEG AND M-JPEG IMAGES.

The network distribution of JPEG and M-JPEG coded images may be affected by the loss of cells containing the coded DCT coefficients generated by this compression mechanism.

As shown in figure 5.5a, the conventional approach to enhancement, image analysis and error concealment is to perform these operations after image reconstruction and in the pixel domain. There are advantages in certain circumstances in carrying out the image processing before reconstruction as is shown in figure 5.5b, and next few chapters discuss several methods of error concealment performed in the transform domain.



For a standard implementation of the JPEG compression algorithm that uses variations of Huffman and variable length coding, the loss of a cell could result in the loss of a whole frame or more since the decoder cannot recover from an

64

error unless additional coding information is inserted in the entropy coded data stream.

An ATM cell may contain several macroblocks (depending upon the information within that part of the image). Therefore to avoid bursts of lost macroblocks it is necessary to interleave the macroblocks before transmission.

It should be noticed that for such applications as a catalogue browsing system in which a collection of individual still images are being transmitted, it is not possible to use information from the previous frame and an error concealment method using information from only within the current frame is required. The object of the research was to find simple methods of error concealment that were acceptable and yet could be executed in real time.

#### 5.7 JPEG AND M-JPEG FILE STRUCTURE.

#### 5.7.1 JPEG File Structure.

The JPEG algorithm has been designed to become a general purpose still picture compression standard that may be used for a broad range of applications, including file distribution. JPEG also specifies a file interchange format that may be used to transfer compressed image files from one application to another.

The syntax for sequential DCT-based, progressive DCT-based and lossless modes of operation is shown in figure 5.6, this is the syntax as recommended in the JPEG compression algorithm.



Marker Codes in JPEG Images.

Compressed image data is described by a uniform structure and a set of parameters for all modes of operation. The various parts of the compressed image data are identified by specific two-byte codes called markers. Some markers are followed by particular sequences of parameters such as table specifications and

headers. Others are used without parameters, such as start-of-image and end-of-image. Table 6.1 is a snapshot of some marker codes specified in the JPEG compression algorithm [2].

Table 6.1 A Sample of Marker Code Assignments				
Code Assignment	Symbol	Description		
FF D8 (Hex)	SOI	Start of Image		
FF D9 (Hex)	EOI	End of Image		
FF DA (Hex)	SOS	Start of Scan		
FF DD (Hex)	DRI	Define Restart Interval		
FF D0 - FF D7 (Hex)	RST	Reset Marker with modulo 8 Count		
FF C4 (Hex)	DHT	Define Huffman Table(s)		
FF DB (Hex)	DQT	Define Quantisation Table(s)		
FF FE (Hex)	СОМ	Comment		
:	:	••		
etc	etc	etc		

The entropy coded data can be segmented and a reset marker used to isolate entropy-coded data segments. The encoder outputs the reset markers, intermixed with the entropy-coded data at regular user defined reset intervals. Reset markers can be identified without having to decode the compressed data as all markers start at a byte boundary. If the entropy-coded data ends before a byte boundary, it is padded to the next byte boundary with the required number of '1' bits. Because the segments can be independently decoded, the segmented mode of operation leads to other application-specific uses, such as parallel encoding and decoding, isolation of data corruption, and semi-random access of entropy-coded segments. The use of reset markers provides the option of cell error concealment and various methods will be investigated.

#### 5.7.2 M-JPEG File Structure.

The image sequences used in this work were captured by the Videologic<sup>TM</sup> compression hardware [39]. Editing, if required, was processed off-line and Videologic<sup>TM</sup> have kindly provided information on the file structure. The file structure for video is in a RIFF (Resources Interchange File Format) format. In a RIFF format, the information is separated into chunks, each chunk consists of a header followed by data.



The first chunk in the image file contains format information such as image size, picture quality and system bandwidth. The next chunk holds the compressed video data. Next is a size chunk which defines the number of data blocks in the data chunk and the duration of the video clip. At the end of the file is an index chunk that stores information such as the position for each of the JPEG images held in the data chunk.

In the video data chunk, each compressed image is stored in a data block and the series of data blocks are stored sequentially to form the complete data chunk. With each compressed image data block, there are associated start and end marker blocks which contain control information. The compressed image data block is a frame which consists of a series of entropy coded segments. The entropy coded segments conform to the JPEG compression algorithm and each consists of a user defined number of macroblocks.

With information on both the structure of the JPEG and M-JPEG algorithms known, error concealment can then be applied to the entropy coded segments to recover the loss of macroblocks during transmission. All concealment techniques considered in the next few chapters are implemented before decompression into the pixel domain.

# CHAPTER 6 ▷▷▷

.



### CHAPTER 6 : ERROR CONCEALMENT ALGORITHMS.

#### 6.1 ERROR CONCEALMENT OF JPEG STILL IMAGES.

The Still Images used in this chapter were standard JPEG compressed images with the 4:2:2  $YC_rC_b$  format. The size of each image was  $512 \times 512$  pixels and a compression ratio of 20:1 was achievable without visually perceptible loss of image quality. Three test images were used for comparison of the various error concealment algorithms, they are :

Image 1 : "Lena", and Image 2 : "Peppers", which are widely used standard images. Image 3 : "Toys", which is a locally composed image with colour and texture.

#### 6.1.1 Replacement by a Fixed Colour Block.

This work commenced by accessing the possibility of simply replacing the lost block with a fixed luminance block or with a block of a suitably chosen colour. A study of the human perception system indicates its sensitivity to luminance contrast, rather than the absolute luminance values themselves. As figure 6.1a shows, the two smaller squares in the middle have equal luminance values but they do not appear equally bright. For a darker surrounding, the same luminance

value will appear brighter than in a lighter surrounding.

For colour images, in addition to contrast illusion, there is the problem of colour sensitivity. Some of the first controlled experiments in colour sensitivity were done by MacAdam [40]. The observer viewed a disk made



Fig 6.1a: Luminance Contrast. Small squares in the middle have equal luminance but do not appear equally bright.

up of two hemispheres of different colours on a neutral background. One colour is fixed and the other could be adjusted by the user. The observer tried to match the fixed colour by controlling the variable colour along a path in tristimulus space. For this work, the luminance was held constant so that only the chromaticity changed.

The results yielded the sensitivity ellipses as shown in figure 6.1b. The ellipses represent a difference in chrominance that is just noticeable.

For a monochrome image, a fixed



luminance block will fit in well only within an area of similar luminance values due to the effect of luminance contrast. For a colour image, it is not just a matter of choosing the luminance value but the chrominance value also needs to be selected. From the colour sensitivity ellipses of figure 6.1b, it is apparent from the small ellipses that a fixed chrominance block will look quite out of place in the wrong colour space, e.g., a red block will look out of place in a green background. Therefore, replacement of the lost block by a fixed luminance or chrominance block requires careful selection of the block characteristics.

#### 6.1.2 Replacement with Previous Macroblock in the Image.

Replacing the lost macroblock with the previous macroblock in the frame is another simple method which does not require a frame store and can easily be

performed in real time. Implementation is above the Segmentation and Reassembly (SAR) Layer and the only storage required is a buffer to store the previous macroblock. Each lost block is replaced with the block in the previous macroblock.;



$$f_L(x_{ii}) = f(x_{pi}), \quad \text{where } i = 1, 2, 3, 4 \qquad \dots (6.1)$$

In this implementation, there is no need to decode the Huffman and Variable Length coded data segment. The whole chunk of entropy coded data between the last two reset markers is selected to replace the lost macroblock. Subjectively, the error concealment appears to be acceptable for relatively small error rates. In certain cases, a lost block is replaced by a block with a large contrast compared with the surrounding area and this replacement would appear distracting in the image.

#### 6.1.3 Replacement using Neighbouring Blocks.

The results achieved for the previous methods can be improved if the content of the neighbouring blocks can be used to estimate the lost block. With the assumption that the transform coefficients between blocks are correlated, simulations were carried out for both linear and quadratic interpolation of the respective corresponding DCT coefficients in the neighbouring blocks.

The two methods defined in the next three sections are algorithms which can be used for correcting single frame JPEG images. Likewise, the error concealment methods mentioned below can be applied even for weakly correlated frames in an image sequence, this is because the algorithms depend only on information within the current frame.

Implementation of the two methods requires partial decoding of the entropy coded data segments. The Huffman and variable length coded data needs to be converted back into the encoded transform domain. For each lost block, an equation will be formed from a weighted sum of the neighbouring blocks and this equation is applied in estimating all the transform coefficients within the lost block.

#### 6.1.4 Interpolation Methods.

This section explores the *divide differences* method of interpolation [41]. The strategy used in approximating unknown values of the function is straightforward. Find a degree-n polynomial that fits the set of data points and assume that the polynomial and the function behave nearly the same over the interval in question. A first degree polynomial leads to the familiar linear interpolation and to approximate functions that are far from linear a higher

degree polynomial would be required. The general degree-n polynomial equation for the method of divide differences is shown in (6.2).

$$f_n(x) = f(x_1) + (x - x_1)f[x_1, x_2] + (x - x_1)(x - x_2)f[x_1, x_2, x_3] + \dots + (x - x_1)(x - x_2)\dots(x - x_{n-1})f[x_1, x_2, \dots, x_n]$$
 ...(6.2)

where,

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}, \ f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}, \ f[x_1, x_2, \dots, x_n] = \frac{f[x_2, x_3, \dots, x_n] - f[x_1, x_2, \dots, x_{n-1}]}{x_n - x_1}$$

#### 6.1.5 Replacement using Linear Interpolation of Neighbouring Blocks. (Degree-1 Polynomial)

The use of the left and right macroblocks are shown in figure 6.3. For the JPEG test images used in this work, a macroblock is defined as an area of  $16 \times 16$ 



pixels. Within this area, there are four  $8 \times 8$  luminance blocks and one subsampled  $8 \times 8$  chrominance block for each  $C_r$  and  $C_b$  component. Each lost component in the YC<sub>r</sub>C<sub>b</sub> format can be estimated from a weighted average of the neighbouring blocks. The general form of a degree-1 interpolation equation is shown here in (6.3).

$$f_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1} (x - x_1) \qquad \dots (6.3)$$

#### Luminance.

The four luminance blocks are separated into top and bottom rows and each row can be estimated separately. Both the lost blocks in each row,  $x_3$  and  $x_4$ , can be predicted with information from the nearest 2 blocks,  $x_2$  and  $x_5$ . Equation (6.3) can be simplified into a weighted sum of the known blocks for each lost block and the resulting equations are as shown below.

$$\begin{aligned} f_{L1}(x_{t3}) &= \frac{2}{3}f(x_{t2}) + \frac{1}{3}f(x_{t5}), \quad f_{L1}(x_{t4}) &= \frac{1}{3}f(x_{t2}) + \frac{2}{3}f(x_{t5}) \\ f_{L1}(x_{b3}) &= \frac{2}{3}f(x_{b2}) + \frac{1}{3}f(x_{b5}), \quad f_{L1}(x_{b4}) &= \frac{1}{3}f(x_{b2}) + \frac{2}{3}f(x_{b5}) \end{aligned}$$

#### Chrominance.

There are two chrominance blocks in each macroblock, one from each  $C_r$  and  $C_b$  component. To form a degree 1 polynomial, there must be at least two known values. Using the left and right macroblock, a weighted sum of the known components can be found for the chrominance component  $C_r$  and  $C_b$ .

$$f_{C1}(x_2) = \frac{1}{2}f(x_1) + \frac{1}{2}f(x_3)$$

#### 6.1.6 Replacement using Quadratic Interpolation of Neighbouring Blocks. (Degree-2 Polynomial)

The general form of a degree-2 interpolation equation is shown in (6.4). This method of *divide differences* [41] uses fewer mathematical operations when compared with the Lagrange interpolating polynomial. Previous calculations can be reused in estimating the higher order portion within the equation.

$$f_2(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) + \frac{f(x_2, x_3) - f(x_1, x_2)}{x_3 - x_1}(x - x_1)(x - x_2) \quad \dots (6.4)$$

A lost macroblock can therefore be reconstructed with information from the neighbouring macroblock. This initial method uses information in one direction only. The left and right macroblock are assumed to be present and as shown in figure 6.3.

#### Luminance.

The four luminance blocks are separated into top and bottom rows. The two blocks in each row can be estimated separately. There are six blocks in each row, the first two blocks  $x_1$  and  $x_2$  from macroblock (*i*-1), blocks  $x_3$  and  $x_4$  from the lost macroblock (*i*) and the last two blocks  $x_5$  and  $x_6$  from macroblock (*i*+1). The lost block  $x_3$  can be predicted with information from the nearest 3 blocks,  $x_1$ ,  $x_2$  and  $x_5$ , similarly  $x_4$  can be predicted from  $x_2$ ,  $x_5$  and  $x_6$ . The equations for estimating the replacement are shown in (6.5).

$$f_{L2}(x_{i4}) = f(x_{i2}) + \frac{1}{3} (f(x_{i5}) - f(x_{i2}))(x_{i5} - x_{i2}) + \left(\frac{f(x_{i6})}{4} - \frac{f(x_{i3})}{12} + \frac{f(x_{i2})}{12}\right)(x_{i4} - x_{i2})(x_{i4} - x_{i5})$$

$$f_{L2}(x_{i3}) = f(x_{i1}) + (f(x_{i2}) - f(x_{i1}))(x_{i3} - x_{i1}) + \left(\frac{f(x_{i5})}{12} - \frac{f(x_{i2})}{3} + \frac{f(x_{i1})}{4}\right)(x_{i3} - x_{i1})(x_{i3} - x_{i2}) \dots (6.5)$$

The element separation is one, and by rearranging the terms the equations can be simplified into a weighted sum of the known blocks.

$$\begin{split} f_{L2}(x_{t3}) &= -\frac{1}{2}f(x_{t1}) + \frac{4}{3}f(x_{t2}) + \frac{1}{6}f(x_{t5}) , \quad f_{L2}(x_{t4}) &= \frac{1}{6}f(x_{t2}) + \frac{4}{3}f(x_{t5}) - \frac{1}{2}f(x_{t6}) \\ f_{L2}(x_{b3}) &= -\frac{1}{2}f(x_{b1}) + \frac{4}{3}f(x_{b2}) + \frac{1}{6}f(x_{b5}) , \quad f_{L2}(x_{b4}) &= \frac{1}{6}f(x_{b2}) + \frac{4}{3}f(x_{b5}) - \frac{1}{2}f(x_{b6}) \end{split}$$

#### Chrominance.

There are two chrominance blocks in each macroblock, one from each  $C_r$  and  $C_b$  component. To form a degree-2 polynomial, there must be at least three known values. For each chrominance component, the lost block  $x_3$  requires information from the nearest three blocks,  $x_I$ ,  $x_2$  and  $x_4$  which are from macroblock (i-2), (i-1) and (i+1) respectively.

$$f_{C2}(x_3) = f(x_1) + (f(x_2) - f(x_1))(x_3 - x_1) + \left(\frac{f(x_4)}{6} - \frac{f(x_2)}{2} + \frac{f(x_1)}{3}\right)(x_3 - x_1)(x_3 - x_2) \quad \dots (6.6)$$

Again, with an element separation of one and by rearranging the terms in the equation above, a simple weighted sum of the known components can be found for the chrominance component  $C_r$  and  $C_b$ .

$$f_{C2}(x_3) = -\frac{1}{3}f(x_1) + f(x_2) + \frac{1}{3}f(x_4)$$

# 6.2 PERFORMANCE OF CONCEALMENT ALGORITHMS FOR JPEG IMAGES.

Results are presented in Colour plate 6.1, 6.2 and 6.3 for *Image 1: "Lena"*, *Image 2: "Peppers"* and *Image 3: "Toys"* respectively. There are a set of four images in each colour plate. In each colour plate, image A is the original image with 5% of the macroblocks lost and each lost block is replaced by a black block. This allows the position of loss to be clearly seen. Image B is generated by replacing the lost blocks with the previous macroblock. Image C show errors replaced with information from linear interpolation of the adjacent blocks before and after the lost block. Image D shows the replacement results of implementing a quadratic interpolation of adjacent blocks.

#### 6.2.1 Image 1: "Lena".

As seen in Image A, replacement by black blocks produces a poor result but it should be noted that without even this level of concealment, decompression of the image might not be possible. Image B is the result of replacement by previous macroblocks. Approximately half of the lost blocks are well concealed. Some concealment errors include the three lost macroblocks on the right cheek being replaced by their respective previous macroblocks which contain part of the hair structure. Similarly, a part of the left cheek was carried forward to the hair. In both cases, the sharp contrast between the replacement block and the surrounding area is obvious and distracting.

Results for Image C shows a big improvement over the first two methods. Replacement blocks take an average of the surrounding left and right blocks and this works well for areas where the surroundings are similar. For contrasting left and right areas, the replacement blocks which are an average of the surrounding, are less objectable then the errors from the earlier method.

Image D replaces the lost blocks with quadratic interpolated data. Quadratic interpolation generates a higher order function and results indicate a tendency to overcompensate the error. For example, an area around the nose was lost and the quadratic interpolated data provided additional information resulting in a replacement block with a nose-like structure.

#### 6.2.2 Image 2: "Peppers".

Replacement by black blocks produces a poor result but at least the image can be displayed. Image B is the result of replacement by previous macroblocks. More than half of the lost blocks are well concealed. Errors are obvious and distracting when the replacement block falls into a contrasting surrounding area.

Replacement blocks for image C use an average of the surrounding left and right blocks and this works well for areas where the surroundings are similar. For contrasting areas, sharp edges around the blocks are not so prominent as the averaging tends to smooth out the edges. Image D replaces the lost blocks with quadratic interpolated data. Result are quite similar to Image C, overcompensated areas within the main green capsicum are not apparent.

#### 6.2.3 Image 3: "Toys".

The original image has a dark background, therefore Image A which replaces the lost block by a fixed black block concealed up to 20% of the lost macroblocks. For image B more than half of the lost blocks are well concealed by previous macroblock replacement. Errors are obvious and distracting when the replacement block falls into a contrasting area.

Image C uses linearly interpolated data for the replacement and this works well for areas where the surroundings are similar. For contrasting areas, sharp edges around the blocks are not so prominent, the averaging tends to smooth out the

# IMAGE 1: "LENA"

JPEG Compressed Still Image



Colour Plate 6.1



IMAGE A 5% Loss of Macroblocks Replacement by Black Block

IMAGE B Replacement by Previous Macroblock in the Image



Replacement using Neighbouring Macroblocks (Linear Interpolation)



IMAGE D Replacement using Neighbouring Macroblocks (Quadratic Interpolation)

Colour Plate 6.2

76

#### IMAGE 3: "TOYS" JPEG Compressed Still Image

S.



Colour Plate 6.3

edges. Image D replaces the lost blocks with quadratic interpolated data. Results are quite similar to Image C, overcompensated areas within the right hand toy are not apparent.

#### 6.3 ERROR CONCEALMENT OF M-JPEG IMAGE SEQUENCES.

Two M-JPEG image sequences were used in this chapter to test the error concealment algorithms, they are:

Image Sequence 1 : "Movie", this is a 25 frames/sec sequence with a duration of approximately 25 secs. There are over 700 individual frames in this sequence and each frame contains  $320 \times 240$  pixels.

Image Sequence 2 : "Shoes", which is a sequence used in a catalogue browsing system. This sequence consists of high quality individual still frames, each frame is of size  $512 \times 288$  pixels and there are a total of 8 frames in this sequence. For this sequence, each image is shown for 3 seconds.

#### 6.3.1 Replacement by Black Macroblock.

This method is easy to implement, the only computation required is in identification of the lost macroblock. Each lost macroblock is substituted by a pre-stored entropy coded data segment representing the fixed colour macroblock. A more detailed description of this concealment mechanism is presented in an earlier section on the same method for JPEG still images.

#### 6.3.2 Replacement by Previous Frame.

This method works for correlated frames only and would require the system to have a frame store. If the incoming video stream is corrupted with any error, the current frame could be dropped and the previous frame displayed. Experimenting with image sequence 1 and a cell loss rate of  $10^{-3}$  results in a image sequence with approximately one frame in error out of every three or four frames.

Subjectively, the sequence looks jerky especially if a few consecutive frames are in error. The overall consensus is that it is satisfactory as long as the error rate is small and the images are correlated. For the same sequence, with an error rate of  $5 \times 10^{-3}$ , there would be on average an error every frame, resulting in nearly no motion being displayed.



Alternatively the block could be replaced by the equivalent block in the previous frame which would allow updating of most of the picture and unless the same block had errors in successive frames, it would not have the jerky effect.

This is an acceptable error concealment method for image sequences with highly correlated successive frames. For sequences with a weak correlation between frames, replacement by the previous frame would not be a suitable method.

#### 6.3.3 Replacement by Previous Macroblock in the same Frame.

This method is the same as that described in an earlier section for JPEG still images. This is a simple method which requires only a buffer to store the previous macroblock.

#### 6.3.4 Replacement using Bilinear Interpolation of Neighbouring Blocks. (Degree-1 Polynomial)

For the M-JPEG test sequences used in this work, a macroblock is defined as an area of  $16 \times 8$  pixels. Within this area, there are two  $8 \times 8$  luminance blocks and one subsampled  $8 \times 8$  chrominance block for each C<sub>r</sub> and C<sub>b</sub> component. Each lost component in the YC<sub>r</sub>C<sub>b</sub> format would be estimated from a weighted sum of the top, bottom, left and right macroblocks. The position and notation of the neighbouring four macroblocks are shown in figure 6.4.



#### Luminance.

The two lost luminance blocks,  $x_{Ll}$  and  $x_{L2}$ , can be estimated with information from the nearest four blocks. The equations required for estimating the lost luminance blocks are as shown below and are a weighted sum of the known blocks.

$$\begin{aligned} f_{L1}(x_{L1}) &= \frac{2}{7}f(x_{t1}) + \frac{2}{7}f(x_{b1}) + \frac{2}{7}f(x_{t2}) + \frac{1}{7}f(x_{t1}) \\ f_{L1}(x_{L2}) &= \frac{2}{7}f(x_{t2}) + \frac{2}{7}f(x_{b2}) + \frac{1}{7}f(x_{l2}) + \frac{2}{7}f(x_{t1}) \end{aligned}$$

Values of all the transform coefficients in the lost block can therefore be estimated from weights of the corresponding transform coefficients in the neighbouring blocks.

#### Chrominance.

There are two chrominance blocks in each macroblock, one from each  $C_r$  and  $C_b$  component. Using the top, bottom, left and right macroblock, a weighted sum of the known components can be found for the chrominance components  $C_r$  and  $C_b$ . All the transform coefficients in each block can thus be estimated from the same equation.

$$f_{Cl}(x_L) = \frac{1}{4}f(x_l) + \frac{1}{4}f(x_b) + \frac{1}{4}f(x_l) + \frac{1}{4}f(x_r)$$

#### 6.4 PERFORMANCE OF CONCEALMENT ALGORITHMS FOR M-JPEG IMAGES.

The performance of the algorithms are shown in Colour Plates 6.4, 6.5, 6.6 and 6.7 for *Image Sequence 1: "Movie"* and Colour Plate 6.8 and 6.9 for *Image Sequence 2: "Shoes"*. Colour plate 6.10 is a comparison between three concealment methods, where each image is a magnification of an area of interest.

#### 6.4.1 Image Sequence 1: "Movie".

With a frame size of  $320 \times 240$  pixels there will be a total of 600 macroblocks in a frame. Assuming a loss rate of  $5 \times 10^{-3}$ , there will be an average of three lost macroblocks in each frame. A snapshot of eight frames are shown for this sequence and each frame is separated by a distance of 40 frames, which is 1.6 secs. Colour plate 6.4 shows where the errors are in each individual frame by colouring the lost areas white. Colour plate 6.5 is the same sequence with the lost macroblocks replaced by black blocks. This sequence has a dark background and some black block replacements fit in well with the background.

Colour plate 6.6 replaces each error with the previous macroblock in the same frame. This works well for concealment in areas within the background or areas with similar structure and colour, up to 50% of the errors can easily be concealed



with this method. Certain concealment errors are very prominent. For example in frame 80 the lost macroblock around the cheek is replaced by the previous macroblock which contains a part of the nostril, this replacement is inappropriate. In frame 200 the macroblock next to the left eye is lost, replacement by the previous macroblock shows a part of the eye. This particular concealment is also very distracting.

The result for replacement by bilinear interpolation of neighbouring macroblocks is shown in colour plate 6.7. This method of replacement achieves a much better result when compared with earlier methods. Errors in the background are well hidden. Error concealment in Frames 80 and 200 is much better than the earlier method, the lost macroblock on the cheek in Frame 80 and the lost macroblock near the eye in Frame 200 are well concealed. Averaging of the surrounding areas will smooth out information within the replacement macroblock. If there is a sharp edge across the macroblock, this method will not be able to detect the edge and use this information to improve on the concealment but the replacement on the whole is satisfactory.

#### 6.4.2 Image Sequence 2: "Shoes".

Image sequence 2 is a series of eight high-quality images for a simulated catalogue browsing system, where each image is displayed for a duration of about 3 seconds. With a frame size of  $512 \times 288$  pixels, there will be a total of 1152 macroblocks in a frame. Assuming a loss rate of  $2 \times 10^{-2}$ , there will be an average of 20 loss macroblocks in each frame.

Colour plate 6.8 shows the sequence with the lost macroblocks replaced by black blocks, it also shows where the errors are within each individual image. For individual frames, black blocks are not very annoying but without replacement, the whole frame would be lost.

Colour plate 6.9 replaces each lost macroblock by bilinear interpolation of neighbouring macroblocks. This method extracts information from the neighbouring blocks and the weighted sum of all the values will be used to represent the lost macroblock. Averaging of the surrounding areas will smooth out information within the replacement macroblock. This method of replacement achieves a much better subjective result when compared with previous methods.

#### 6.4.3 Comparsion between Methods.

Comparison of the error concealment methods: Method 1: replacement by black macroblock, Method 2: replacement by previous macroblock and Method 3:

replacement by linear interpolation of neighbouring macroblock are shown by way of selected examples in colour plate 6.10. A portion of the image is magnified and this area is filled with errors that can be difficult to conceal.

In the first set of images the errors are around the right cheek and a portion of the lips. Method 2 carries the previous macroblock forward and it did not work well as part of the turquoise blue background is used as the replacement. Method 3 works quite well for this image, the lost macroblocks are replaced with skin-coloured macroblocks. But a portion of the lips cannot easily be replaced by concealment as details are localised within one or two macroblocks.

The second set of images have errors near the nose and the eye. Method 2 provided poor replacements, the portion of the eye and nostril that are used for replacement can be very unpleasant. Method 3 performed extremely well and the method of bilinear interpolation senses the importance of the other surrounding macroblocks and the bilinear interpolated replacement fits in well with the rest of the image.

The third set of images have errors at the edge of the face and the left cheek. Method 2 brought the background into the face and the cheek is replaced with a macroblock that is brighter than the surrounding area. Method 3 corrected well for the error on the cheek but a slight tint of skin colour is present in part of the replacement macroblock that is supposed to be in the turquoise background. The edge of the face that separated the background is discontinued at the area of the replacement block but is not too annoying.

The last set of images magnifies the area of the shoulder, three macroblocks are lost, two of them are adjacent to each other and the other is between the shoulder and the girl's blouse. Method 2 replaces one of the macroblocks quite well. Method 3 did not work very well for the two consecutive macroblocks that are lost. This is because of the limited information available when consecutive macroblocks are lost and it also stresses the importance of macroblock interleaving to prevent a continuous burst of lost macroblocks.

#### 6.5 CONCLUSION AND FURTHER IMPROVEMENTS.

A simple replacement by a fixed colour block would be acceptable for very low cell loss rate. For a higher cell loss rate, the number of black block replacements would increase. Most of the black blocks would be in different positions for each frame, for an image sequence with 25 frames per second, the change in positions would give an illusion that the blocks are jumping around. The human perception



system tends to track areas that are moving and the moving black blocks are very distracting. For *image sequence 2*, the black blocks are stationary and they disappear when the frame is changed and would not appear to be as annoying.

The second method of previous frame replacement is very satisfactory but requires that successive frames are correlated. In a moving sequence this is normally the case unless there is a lot of motion or a change of scene but for applications where the individual frames are completely different, e.g., in a catalogue browsing application, this method does not work.

The third method of previous macroblock replacement is for moving sequences with uncorrelated frames or still images. This is a simple method that requires little computation, only slight buffering being required to store the previous block. It works well around areas which are similar, for example the background of a scene, but not when there is a sharp transition of the pixel intensities between the replacement block and the surrounding area. This method is still an improvement over the first two methods.

The fourth method of linear/bilinear interpolation using neighbouring blocks produces satisfactory results for many image types but does require some extra computation at the receiver. The processing was implemented in the transform domain since this enabled advantage to be taken of the energy concentration properties of the DCT. Linear interpolation which uses the average of the values in the neighbouring blocks gives the best results. This method achieved a better subjective result than duplicating the previous block to replace the lost block.

The fifth method of quadratic interpolation using neighbouring blocks requires some extra computation at the receiver. This method assumes a quadratic fit between the known and lost blocks and it tends to overcompensate for the errors. The replacement was used only for JPEG still images.

The work concentrated on the Method of Linear/Bilinear Interpolation using Neighbouring Blocks. A study of the distribution of pixel and transform coefficients was then conducted and is discussed in the next chapter where the assumption that there exists correlation between the transform coefficients is verified. The results from these investigations should provide a better understanding of the image characteristics and hopefully help to improve on the performance of the algorithm.



Colour Plate 6.4

游

## IMAGE SEQUENCE 1: "MOVIE" Replacement by Black Macroblock













\*



FRAME 120





Colour Plate 6.5

85

# IMAGE SEQUENCE 1: "MOVIE" Replacement by Previous Macroblock











瓣

FRAME 40



FRAME 120





Colour Plate 6.6

86





Colour Plate 6.7



\*

Colour Plate 6.8




Colour Plate 6.9

\*

METHOD 1	METHOD 2	METHOD 3
METHOD 1	METHOD 2	METHOD 3
METHOD 1	METHOD 2	METHOD 3
METHOD 1	METHOD 2	METHOD 3
METHOD 1 Replacement by Black Macroblock	METHOD 2 Replacement by Previous Macroblock	METHOD 3 Replacement by Linear Interpolation of Neighbouring Macroblock

Colour Plate 6.10

# CHAPTER 7 ▷▷▷

•



### CHAPTER 7 : STATISTICAL MEASUREMENTS OF IMAGES.

This chapter investigates the distribution of the pixels in a digital image and the distributions of the transform coefficients in a DCT based compressed image. The second part of this chapter looks into the modelling of the transform coefficients as a Gaussian or Laplacian distribution and one section analyses the 1-dimensional correlation for the transform coefficients within an image. This study provided an interesting insight into the DCT and how the properties of the DC and AC transform coefficients differ. This information in turn is used by a later chapter to enhance a proposed cell loss concealment algorithm.

### 7.1 TEST IMAGES USED FOR STATISTICAL MEASUREMENTS.

Four images were used for the understanding of pixel and transform coefficient distributions, they are as shown in Colour Plate 7.1 and described below :

Image A : "Lena" -	A head and shoulders image. One of the widely
	used standard images.
Image B : "Peppers" -	The whole image is covered by numerous green
	and red capsicums and chilli pods. Another one of
	the widely used standard images.
Image C : "Toys" -	A locally composed image of a toy with coloured
	rings and another toy with colour and texture.
Image D : "Random" -	An image in which each pixel is an independent
	sample of a random variable.

Image A: "Lena" and Image B: "Peppers" are standard images with a resolution of  $512 \times 512$  pixels each of 24 bits, with 8 bits for each of the R, G and B colour

### IMAGES FOR STATISTICAL MEASUREMENTS

JPEG Compressed Still Images



Colour Plate 7.1

A STATE

components, giving each component 256 possible discrete levels. Image C: "Toys" is a locally composed image, it was captured using a digitisation board at a resolution of  $512 \times 512$  pixels and 24 bits depth. Image D: "Random" is generated from a computer program and the value for each component is an independent sample obtained from a uniformly distributed random generator.

To generate the transform coefficients, each image is divided into a series of  $8 \times 8$  pixel blocks and each block is transformed from the spatial domain into the transform domain using the DCT (Discrete Cosine Transform). After transformation, each transform block will consist of 64 coefficients. The transform coefficients of each block are then quantised according to the human visual compensated matrix, where each coefficient is quantised by a unique factor [2]. Due to the high energy packing efficiency of the DCT, most of the energy will be concentrated in the top left corner of the transformed block, e.g., up to about 20% of the total AC energy is present in each of the first order AC coefficients F(0,1) and F(1,0) [Chap 9]. For this reason the study concentrates on only the first four transform coefficients, i.e., the DC coefficient and the first three AC coefficients F(0,1), F(1,0) and F(1,1).

### 7.2 DISTRIBUTION OF THE PIXELS & TRANSFORM COEFFICIENTS.

A simple and useful tool in image processing is the histogram. This function summarises the distribution of pixels or transform coefficients in a two dimensional plot. where the abscissa is intensity level and the ordinate is the frequency of occurance. This histogram is a first-order approximation of the probability density function (pdf).

### 7.2.1 Histograms of Test Image A "Lena".

Figure 7.1a shows the histogram of test image "Lena". This is a well distributed pixel distribution with no intensity clipping on either side of the intensity. From the pdf of the "Lena" image, it is difficult to make assumptions about the statistical characteristics of the distribution and modelling of the pixel distribution other than a uniform pdf seem to be unproductive. The distribution is clearly not unimodal over the intensity interval and it is hardly surprising that the pdf is not a bell-shaped Gaussian distribution.



Most natural images consist of a background (usually with a constant range of pixel intensities) and an area of interest which frequently stands out from the rest of the image (background). Therefore, it is quite common to have two main peaks in the histogram of the image. The two peaks are usually separated by a significant range of intensities.

The distribution of the DC coefficients F(0,0) is presented as a histogram in Figure 7.1b. Based on a block size of  $8 \times 8$  pixels. The "Lena" image of  $512 \times 512$  pixels will generate a total of  $64 \times 64$  DC coefficients after transformation. For transformations such as DCT, the DC coefficient is a scaled average of the pixel values within the  $8 \times 8$  block. From the histogram it is observed that the profile of the pixel distribution is carried over to the DC transformed coefficients. Both the pdf of the DC coefficients and the pdf of the pixel intensities do not exhibit any well-defined statistical characteristics.

The measured value of the image mean is -2.0 and the standard deviation is 22.4. The maximum and minimum DC luminance coefficients generated after transformation and quantisation is  $\pm$  64. It is obvious from the histogram that the image does not exhibit a bell-shaped distribution and it is not symmetrical about the mean. Therefore an approximation to a Gaussian distribution is not rational.

A new distribution can be formed from a linear summation of the samples from any distribution (with finite mean and variance) with independent and identically distributed random variables and according to the Central Limit Theorem, the new distribution will converge to a standard Gaussian (normal) distribution if there is a large enough number of samples.

As the DC coefficient is formed from a linear operation on the  $8 \times 8$  pixel block, the new distribution would be expected to approximate to a normal distribution. However, as the pixels are highly correlated, and therefore dependent random



Since the introduction of Transform Coding and its extensive acceptance as a competent coding algorithm, there have been a few studies on the distribution of the AC coefficients[10,42]. The widely expressed opinions regarding the distributions are that they are either of a Gaussian, or a Gamma (Laplacian, is a special case of Gamma) nature.

The AC Transform Coefficients are formed from a cosinusoidal combination of the pixels in the  $8 \times 8$  blocks. and the Orthogonality criterion of the DCT implies that product of any two independent AC basis vectors will have zero mean values. A typical AC coefficient distribution will be approximately symmetrical about zero [10].

The AC coefficient F(0,1) has the distribution shown in Figure 7.1c. The distribution of the AC coefficients is interesting in the statistical sense.

The measured mean of the distribution is zero and the standard deviation is 8.3. The discrete value of the coefficients are joined by straight lines and a transition from a monotonically incrementing/decrementing interval will result in a sharp knife edge peak/groove rather than a smooth transition. This helps to explain the sharp peak around the zero interval.

All the transform coefficients are quantised by a table based on the psychovisual threshold of the human vision system. For the transform coefficient F(0,1), the normalisation factor is 11, therefore any value between 5.5 and -5.5 will be rounded up to zero. This quantisation will also have a slight effect on the number of zero coefficients.

Figure 7.1d is the histogram of transform coefficient F(1,0). The measured mean is zero and the standard deviation of the distribution is 4.4. A Gaussian Distribution will have 99.7% of its distribution within the range of  $+3\sigma$  and  $-3\sigma$ .



and for this distribution, 99.7% of the data should be within the interval +13 and -13. From the histogram, it seems reasonable to state that the distribution tails off after three standard deviations. The distribution does tend to have a sharp peak which rapidly falls off as the amplitude decreases or increases about the mean value.



For figure 7.1e, the measured standard deviation is 3.1 and the mean is zero. From the histogram, it can be seen that the distribution of the transform coefficient F(1,1) has quite similar properties to the first two transform coefficients mentioned above. The main observation is that the distribution has a sharp peak around the mean value which falls off rapidly.

### 7.2.2 Histograms of Test Image B : "Peppers".

From the test image in colour plate 7.1b, it can be seen that the whole image consists of capsicums and chilli pods. Therefore the two main colours on the image are green and red and this is reflected in the histogram of the pixel intensities (Fig 7.2a). There are two main peaks centred around luminance value 40 and 195 which corresponds to the two main colours red and green respectively. The rest of the luminance values are quite evenly distributed.

Figure 7.2b shows the distribution of the DC transform coefficient F(0,0). For transform coding, this is the mean of the  $8 \times 8$  pixels block. Examining test image A, it is noticed that the profile of the pixel intensities is carried over to the DC transform coefficient distribution, which is hardly surprising if the pixels are highly correlated and the DC coefficients have a high energy content. For test image B, the profile is not carried over from the pixel domain, this is because the DC coefficients do not contain a significantly high amount of the total energy in the  $8 \times 8$  pixels block.

The AC transform coefficients F(0,1), F(1,0) and F(1,1) are represented in figure 7.2c, 7.2d and 7.2e respectively. It is quite obvious that each distribution is exponential in nature and is roughly Gaussian in nature, with a highly peaked mean. A later section will examine how well the AC coefficient fits a Gaussian distribution.



#### 7.2.3 Histograms of Test Image C : "Toys".

Figure 7.3a shows the histogram of test image "Toys". It is a plot of the number of occurrences for each intensity level within the image. The last 30 to 40 lines of the "Toys" image are confined to a narrow range of luminance value in the vicinity of black and this arrangement of the lower region is reflected in the histogram by a sharp peak of the luminance value in the black region. From the histogram, it is quite obvious that the mean of the distribution is below the midrange value. This implies that the image as a whole consists of more pixels which are darker in intensity.

The histogram of the DC coefficient F(0,0) (fig 7.3b) reflects the histogram of the pixel intensity distribution and this is due to the high correlation in the pixel domain which is carried over to the DC transform coefficients.

Figure 7.3c, 7.3d and 7.3e are the Histogram of the AC Coefficient F(0,1), F(1,0) and F(1,1) respectively. The histograms are highly peaked with mean values around zero. A later section will test the goodness of fit in modelling the distributions to fit a Gaussian or Laplacian model.

### 7.2.4 Histograms of Test Image D : "Random".

The pixels of test image D are generated by a uniformly distributed random variable. They are uncorrelated random variables with a value between 0 and 255. Test image D (colour plate 7.1d) looks like an evenly distributed mass of noise-like structures, typical of images from a broadcasting station after transmission hours (with random colour instead of just black and white patches). If the distribution is uniformly distributed, each luminance value will have approximately occurred ( $512 \times 512$ ) / 256 [(total resolution)/luminance levels] times. Figure 7.4a is the distribution of the pixel luminance values and it is quite evenly distributed with a measured mean number of occurrence around 1024, which is the calculated frequency of occurrence.

Figure 7.4b is the DC coefficient F(0,0) distribution. It is a bell-shaped distribution and will be shown later, to resemble a Gaussian distribution. This supports the claim of the Central Limit Theorem and shows that uncorrelated, independent pixels will generate DC coefficients with a Gaussian distribution.

This also proved that natural images (Test Images A, B and C) have highly correlated pixels and the correlation is so strong that the mean (DC coefficients) of the pixels are still correlated and the profile of the DC coefficients does not tend to a Gaussian distribution.



Figure 7.2 : Image B "Peppers". Fig 7.2a:- Histogram of the Pixel Intensities. Fig 7.2b:- Distribution of the DC Transform Coefficient. F(0,0)

86

CHAPTER 7: STATISTICAL MEASUREMENTS OF IMAGES



#### Figure 7.3 : Image C \*Toys". Fig 7.3a:- Histogram of the Pixel Intensities. Fig 7.3b:- Distribution of the DC Transform Coefficient. F(0,0)

66

CHAPTER 7: STATISTICAL MEASUREMENTS OF IMAGES





Values of Coefficients

400

350

300

250

200

150

100

50

-15

Values of Coefficients

Frequency

د

14,6

15

Values of Coefficients

Figure 7.4c, 7.4d and 7.4e are the distributions of the AC coefficients F(0,1), F(1,0) and F(1,1) respectively. The distributions for this image have a larger standard deviation and the shape of the distribution will be spread further out over the intensity range. It is quite obvious from the histogram that the distribution is more Gaussian as compared with Images A, B and C. A later section will investigate this further.

## 7.3 MODELLING OF THE TRANSFORM COEFFICIENT DISTRIBUTIONS.

From the histograms of the three natural images, it is reasonable to assume that the DC coefficients of natural images cannot be accurately modelled as any known mathematical distribution. The nature of the distribution is dependent upon the pixels, whereas the AC transform coefficients are more predictable and it would be possible to check how closely they resembles a known distribution such as the Gaussian or two-sided Laplacian distribution.

### 7.3.1 Linear Regression.

This section will introduce simple linear regression and its use as a statistical tool. A simple linear regression equation takes the form Y = a X + b, where the parameters a and b of the equation are generally unknown but can be estimated by the widely used method of *least squares*.

To plot a line of best fit of Y on X, assume that the X values are fixed and measured without error and the Y values are subjected to random variation. If the Y values do not fall precisely on any straight line, we can select from the family of all possible straight lines the one which gives a "best fit" to the experimental data. This line is called the *regression line* of Y on X, and the equation is called a *regression equation*. After the regression equation has been determined, the strength of the linear relationship can be seen from how well the observed values are predicted by the equation. The objective measure is  $R^2$ , which is the percentage of variation and is as defined below :

$$R^{2} = \frac{\sum (observed values of Y - \overline{Y})^{2}}{\sum (predicted values of Y - \overline{Y})^{2}} \dots (7.1)$$

 $R^2$  is also known as the coefficient of correlation and by definition, the range is  $0 \le R^2 \le 1$ .



### 7.3.2 Gaussian Distribution.

Historically, the Gaussian (Normal) distribution has played a central role in the development of probability and statistics, the Gaussian distribution is popular because it is mathematically well understood. A Gaussian density has the familiar bell-shaped curve and its equation is as shown below.

Gaussian Distribution : 
$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \dots (7.2)$$

where  $\mu$  is the mean and  $\sigma^2$  is the variance

The suitability of the Gaussian distribution as a model for the transform coefficients depends upon the extent to which it fits the observed data. To test the observed data, the function of the Gaussian Distribution can be rearranged as follows to fit a linear equation;

Taking natural logarithm on both sides of (7.2),

$$\ln[f(x)] = \ln\left[\frac{1}{\sqrt{2\pi\sigma}}\right] + -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2$$

if mean  $\mu~=~0$  and rearranging ,

$$\ln[f(x)] = -\frac{1}{2\sigma^2}x^2 + \ln\left[\frac{1}{\sqrt{2\pi\sigma}}\right] \qquad ... (7.3)$$

₿

Linear Equation,

Y = aX + b

From the above, It can be seen that the equation of the Gaussian Distribution can be rearranged into the general form of the equation for a straight line. ln[f(x)] can be plotted as the y-axis and  $x^2$  as the x-axis of a straight line. The aim of this exercise is to examine the distributions of the DC and AC coefficients of test images A, B, C and D for Gaussian behaviour.



Figure 7.5a is a plot of the measured DC coefficients for Image A "Lena" The distribution of the DC coefficients is presented in a different form and the straight line that best fits the observed values is calculated. The calculated regression equation using the method of least squares is  $-0.001x^2 - 4.17$ . As can be seem from figure 7.5a, the observed values vary quite substantially from both sides of the regression equation and this is reflected in a low correlation coefficient of 0.51. Modelling of the DC coefficient as a Gaussian distribution is not sensible.

Similar methods of linear regression were used on the AC coefficients of "Lena" (Image A). The regression line that best fits the observed AC coefficients F(0,1) of Image A is  $-0.004x^2 - 3.98$  and the measured correlation coefficient is 0.74. The regression line that best fits the AC coefficients F(1,0) is  $-0.01x^2 - 3.88$  and its correlation is 0.75.

From both figure 7.5b and 7.5c, it is apparent that the measured values do not vary substantially from the regression line except for values near zero and at the extremity.





Figure 7.5d is the plot for AC coefficients F(1,1) the results are quite similar to AC coefficient F(0,1) and F(1,0). The regression equation that best fits the measured data is  $-0.015x^2 - 3.68$  and the correlation coefficient is measured as 0.736. From all the plots of the AC coefficients, the observed data fits quite closely to the predicted regression equation except near the two extremes. The peakiness of the distribution near zero is partially due to the quantization of the coefficients after transformation, and the lack of that particular AC information is the reason for the high number of zeros. The long tail-off for distributions of AC coefficients is due to the high AC energy in small areas within an image. This will result in quite large negative natural logarithmic values which are difficult to predict and will undermine the regression equation, thus lowering the correlation coefficient. Results of the study for Images B, C and D are presented in figures 7.6 to 7.8 and a summary of the correlation coefficients are presented in table 7.1.

Table 7.1: Coefficient of Correlation $(R^2)$ for Gaussian Model					
	DC coeff. F(0,0)	AC coeff. F(0,1)	AC coeff. F(1.0)	AC coeff. $F(1,1)$	
Image A "Lena"	0.510	0.737	0.750	0.736	
Image B "Peppers"	0.477	0.669	0.690	0.786	
Image C "Toys"	0.015	0.618	0.600	0.514	
Image D "Random"	0.978	0.977	0.983	0.984	



СК.



106

CHAPTER 7: STATISTICAL MEASUREMENTS OF IMAGES



107

The plots for Image D "*Random*" justify the claims of the central limit theorem, the DC and all three AC coefficients behave like Gaussian distributions and this is reflected in the correlation coefficients, which were all above 0.97.

It also proved that the DC coefficients of natural images are highly correlated and therefore dependent. For this reason, the DC distribution of most, if not all, natural images will not tend to a Gaussian distribution.

The AC coefficients of the other test images are less correlated and this is mirrored in the zero mean peaky distributions. The average values of the correlation coefficients is 0.7, therefore modelling of the AC coefficient distributions as Gaussian is only satisfactory. The main disagreements are the peakiness of the distribution and the long tail off.

### 7.3.2 Laplacian Distribution.

The equation of a two-sided Laplacian Distribution is shown below .

Laplacian Distribution :  $f(x) = \frac{c}{2} \exp^{-c|x-d|}$  ... (7.4)

where d is the mean and  $2/c^2$  is the variance

The Laplacian distribution has an exponential curve and the suitability of the Laplacian distribution as a model for the transform coefficients depends on the extent to which it fits the observed data. To test the observed data, the function of the Laplacian Distribution can be rearranged as follows;

Taking natural logarithm on both sides of (7.4),

$$\ln[f(x)] = \ln\left[\frac{c}{2}\right] + -c|x-d|$$

if mean d = 0 and rearranging,

$$\ln[f(x)] = -c|x| + \ln\left[\frac{c}{2}\right] \qquad ... (7.5)$$

Linear Equation,

$$Y = aX + b$$

î

The final equation (7.5) can be plotted as a straight line with ln[f(x)] as the y-axis and |x| as the x-axis. The histograms plotted in section 7.2 can be rearranged to fit a straight line plot by taking the natural logarithm of f(x), where f(x) is the normalised value of the frequency of occurrences for the y-axis and using the absolute value of x (pixel or coefficient intensity) as the x-axis.



The aim of this exercise is to examine the distributions of the DC and AC coefficients of test images A, B, C and D for Laplacian behaviour. Using the method of least squares, a linear regression line can be estimated from the available data and the percentage of variation  $R^2$  can be calculated to give an indication of the goodness of fit.

The distribution of the DC coefficients for "Lena" (Image A) is as shown in figure 7.9a and the regression equation is -0.03|x| - 3.92, with a coefficient of correlation of 0.49. It can be seen that the large variation of the measured data from the best fit regression line and low correlation coefficient indicates that the distribution of the DC coefficients does not follow a Laplacian distribution.

For the AC coefficients F(0,1) (fig 7.9b), the data fits a regression line of -0.13|x|-3.10, with the variation of the data from the regression line measured at 0.89. Figure 7.9c, a plot of AC coefficient F(1,0), has a regression line of -0.236|x|-2.75, the coefficient of correlation being measured at 0.90. For AC coefficient F(1,1) (fig 6.10d), -0.36|x|-2.34 is the best regression line estimated from the method of least squares and the percentage of variation is 0.90. From the plots of the AC coefficients F(0,1), F(1,0) and F(1,1), it is apparent that the variation between the predicted values and the observed values is small and this was reflected in the high correlation coefficients measured. A two-sided Laplacian distribution seems like a good representation of the AC coefficients of "Lena" (Image A). The plots for Image B, C and D are presented in figure 7.10 to 7.12 and the values of the correlation coefficients are tabulated in table 7.2.





Table 7.2: Coefficient of Correlation $(R^2)$ for Laplacian Model					
	DC coeff. F(0.0)	AC coeff. F(0,1)	AC coeff. F(1.0)	AC coeff. F(1.1)	
Image A "Lena"	0.490	0.892	0.900	0.904	
Image B "Peppers"	0.417	0.841	0.858	0.937	
Image C "Toys"	0.024	0.805	0.763	0.741	
Image D "Random"	0.928	0.919	0.937	0.928	

From figure 7.9 to 7.12 and table 7.2, it is possible to reach some conclusions. The DC coefficients for natural images do not tend towards a Laplacian distribution due to the high correlation between neighbouring coefficients, which will generate a distribution which mirror the distribution of the pixels. However,



CHAPTER 7: STATISTICAL MEASUREMENTS OF IMAGES.

-



CHAPTER 7: STATISTICAL MEASUREMENTS OF IMAGES





-2 . -3 ? 4 [(x)] u -5 •6 -7 Coefficient of Correlation  $(R^2) = 0.928$  Regression Line = -0.422\*|x| - 1.360Image D "Random" AC Coefficient F(1,1) 2 n -1 -7 · · · · · -1 **\_\_**0^ ۲ [(x)]] u -4 ۲ •5 -6 •7 -8

absolute (x)

2

n

٥

-1

-9

113

CHAPTER 7: STATISTICAL MEASUREMENTS OF IMAGES

÷.

the AC coefficients F(0,1), F(1,0) and F(1,1) of natural images will have a near symmetrical and zero-mean distribution which can be modelled reasonably well with a Laplacian distribution. The average correlation coefficient of the AC transform coefficients is 0.85, which is very high.

### 7.4 CORRELATION OF TRANSFORM COEFFICIENTS.

Test images A - "Lena", B - "Peppers" and C - "Toys" were used to show how the nature of images affects the 1 dimensional correlations of the transform coefficients. All the images were standard JPEG images, compressed using the JPEG Still Picture Compression Algorithm.

The quantisation operation after the DCT would remove many of the coefficients which is why, as in the earlier sections, only the first four transform coefficients are considered in this study. The physical interpretation of the DCT is that the spacial image is made up of a number of basis images and the coefficients of these basis functions are the values in the frequency domain. The basis images for the first four frequency components are shown in figure 7.13.

For the DC coefficient F(0,0) the basis image has a constant illumination over the block as is shown in figure 7.13a. In the transform domain, the DC coefficient F(0,0) represents the mean value of the  $8 \times 8$  DCT block.

The basis image for the AC coefficient F(0,1) is shown in figure 7.13b. It can be seen that the pixel intensity varies from peak white to peak black in a cosine fashion in the horizontal direction but is constant in the vertical direction. A positive value of the coefficient means that in the spatial domain, the intensity of the pixels decreases cosinusoidally



when moving from left to right whereas a negative value implies that the pixel intensity increases moving from left to right. The basis image for the AC coefficient F(1,0) is shown in figure 7.13c and it can be seen to be similar to the AC coefficient F(0,1) but rotated through 90 degrees. A positive value means that in the spatial domain, the intensity decreases cosinusoidally from top to bottom and a negative value implies that the intensity increases moving from top to bottom.



Colour Plate 7.2

Finally the basis image for the AC coefficient F(1,1) is shown in figure 7.13d. It can be seen to have a half cycle variation of the pixel intensities in both the horizontal and vertical direction. A positive value will have the shade pattern as shown in the basis image and a negative value has a structure that is the opposite of the basis image, i.e., the dark area will be lighter and vice versa.

Each of the coefficients resulting from the DCT operation were then gathered into a 2 dimensional array. so that, in total, there will be 64 arrays holding  $64 \times 64$ values of each of the luminance transform coefficients and  $32 \times 32$  values for the chrominance transform coefficients, (image size of  $512 \times 512$  with YC<sub>b</sub>C<sub>r</sub> 4:2:2 format). Colour plate 7.2 presents typical values of the first four DCT coefficients for the luminance component in the form of a 3-dimensional contour map.

### 7.4.1 Correlation in Test Image A : "Lena".

In the transform domain, the DC coefficient F(0,0) contains the mean value of the  $8 \times 8$  DCT block. It should be noted that because of the subtraction of 128 before carrying out the DCT this is not the same as the mean of the pixel intensities. The correlation between pixels in the image domain for a typical head and shoulders image is very high, e.g., a typical 1-step correlation is about 0.9 and for a lag of 8 picture pixels the correlation is still about 0.7 [10]. With this knowledge of the correlation in the pixel domain, the correlation of the DC coefficients in the transform domain will be expected to be significant.

An image of size  $512 \times 512$  when transformed with a  $8 \times 8$  2-dimensional DCT will generate a total of 4096 DC coefficients. A typical line of the luminance DC coefficients is as shown above in Figure 7.14. The DC coefficients' distribution has a mean of -1.97 and a standard deviation of about 21.8.

The DC coefficients F(0,0) which are the mean values of the  $8 \times 8$  blocks, are estimated to be separated by a distance of about 8 pixels in the spatial domain and the anticipated correlation between adjacent DC coefficients would be expected to be between 0.7 and 0.9.



### Luminance DC Coefficients.

As can be seen from Figure 7.18a, which is the correlogram of the luminance DC coefficients, the correlation between successive DC coefficients of the neighbouring blocks is still quite significant for small block separation.

For a lag of one element, the correlation between adjacent luminance DC coefficients vertically is around 0.89. The horizontally separated luminance DC coefficients have a 1-step correlation of 0.79 and finally the diagonally separated luminance DC coefficients still have a 1-step correlation coefficient of 0.7.

It can be seen that the correlation is slightly greater in the vertical direction, which is probably due partly to the choice of sampling frequency and partly to the fact that there is more vertical structure in the image. Diagonal correlation, as expected, is less than that along either the horizontal or vertical direction, mainly because the spacial separation between blocks is greater than in either the horizontal or vertical direction.

As has been stated in appendix C the normalised auto-correlation for a stationary first order Markov process is given by  $C(k) = \rho^k$ .

The first order Markov process can be plotted as a straight line, the ordinate is the logarithm of the correlation coefficient and the abscissa is the lag or element separation. In the vertical direction, a first order Markov approximation of 0.85 for the DC coefficient is quite a good fit. In the horizontal direction, it can be modelled by  $\rho = 0.70$  and in the diagonal direction, a coefficient of 0.55.

A comparison of the correlation coefficients with those for a Markov process are shown in figure 7.15 and it would seem from this evidence that a first order Markov process would be a good model for the luminance DC coefficients of test image A "Lena".



### Chrominance DC Coefficients.

An image of size  $512 \times 512$  will be downsampled to  $256 \times 256$  for the two chrominance signals (C<sub>r</sub> and C<sub>b</sub> components). The  $8 \times 8$  2-dimensional DCT will generate a total of 1024 chrominance DC coefficients for each chrominance component. The DC coefficients' distribution for the C<sub>b</sub> chrominance value has a mean of -4.96 and a standard deviation of about 5.56. The distribution for the C<sub>r</sub> chrominance value has a mean of 18.87 and a standard deviation of 5.1.

From the "Lena" test image, it can be seen that the image is strongly tinted with a light skin coloured brown. The value of the chrominance DC coefficients should be quite similar and a high inter-element correlation can be expected. Figure 7.16 shows a typical line of chrominance DC coefficients and, as expected, the differences in amplitude between successive chrominance DC coefficients are quite small.



Figure 7.18b and 7.18c show the correlograms of the  $C_r$  and  $C_b$  chrominance DC components respectively. Since the chrominance values are downsampled by a factor of two the chrominance DC coefficients are the mean of an area of  $16 \times 16$  picture pixels.

The inter-element chrominance correlations would be expected to be lower than the similar correlations for the luminance values, as the spacial distance between successive blocks is greater for the chrominance values. However the correlations between chrominance DC coefficients are in fact quite good and compare well with those obtained for the luminance DC coefficients.

### AC Coefficients.

"Lena" AC Transform Coefficients: The AC coefficients studied are the coefficient F(0,1), coefficient F(1,0) and coefficient F(1,1). A typical line of AC coefficient is as shown in Figure 7.17. The values seem to be more random and a lower correlation, if any, would be expected.



"Lena" AC F(0,1) Coefficient: Statistics of the luminance and both the C<sub>r</sub> and C<sub>b</sub> chrominance AC coefficient F(0,1) are as shown in Figure 7.19.

There is a reasonable correlation for the AC coefficient F(0,1) in the vertical direction. It seems quite obvious from the basis image that correlation can only be expected in the vertical direction. There is a continuity in the pixel intensities along the edge when two F(0,1) basis images are placed side by side in the vertical direction and only in the vertical direction.

Therefore, as expected, the horizontal and diagonal direction will not have much correlation.

"Lena" AC F(1,0) Coefficient: Statistics of the luminance and both the C<sub>r</sub> and C<sub>b</sub> chrominance values AC coefficient (1,0) are as shown in Figure 7.20.

The basis image for AC coefficient F(1,0) is shown in Figure 7.13 and it seems quite obvious from the basis image that correlation could be expected in the horizontal direction only. There is a continuity in the pixel intensities along the edge when two F(1,0) basis images are placed side by side in the horizontal direction and only in the horizontal direction. From the correlograms, it can be seen that for this image, there is no significant correlation even in the horizontal direction and as expected, the vertical and diagonal direction do not have much correlation. Wavelet transform theory can be used to decipher information within an image into its various classes of frequency components and the result for image "Lena" shows that the traces of 1-dimensional fundamental frequencies in the horizontal direction are scarce and lacking.

"Lena" AC F(1,1) Coefficient: Statistics of the luminance and both the C<sub>r</sub> and C<sub>b</sub> chrominance values are shown in Figure 7.21. The basis image for AC coefficient F(1,1) is shown in Figure 7.13 and it seems quite obvious from the basis image that there will be very little correlation in any direction. There is no continuity in the pixel intensities along the edge when two F(1,1) basis images are placed side by side in any direction. From the correlograms, it can be seen that as expected for this image, there is no significant correlation in any direction.







This test image of capsicums consists mainly of two prominent colours, red and green. For this study, only the first four transform coefficients were considered.

### DC Coefficients.

"Peppers" Luminance DC Coefficient: A typical line of the luminance DC coefficients is as shown in Figure 7.22. The distribution has a mean of -5.69 and a standard deviation of about 24.1.

The coefficients vary slightly more frequently from positive to negative values than for the previous image due to numerous green and red capsicums in the image. Figure 7.26a is the correlogram of the luminance DC coefficients. For a lag of one, the correlation between adjacent coefficients in the vertical direction is 0.87 and the adjacent coefficients in the horizontal direction have a one step correlation of 0.83. Finally the correlation of the coefficients in the diagonal direction is 0.70.

As expected the diagonal correlation is lower because of the larger physical block separation., whereas the correlations are much stronger in the horizontal and vertical direction.



122

Figure 7.23 is a plot of the k step correlation in the horizontal direction and the logarithm of the correlation coefficient in the vertical direction. The values of the correlations for image "*Peppers*" were plotted as discrete values and straight lines were plotted for the values of first order Markov processes that regress close to the discrete values.

A first order Markov process with  $\rho = 0.80$  is a good fit for the vertical correlations and  $\rho = 0.75$  is a good representation for the horizontal correlations. For the diagonal, the correlation closely follows a first order Markov process of 0.6. From figure 6.25 and the above results, it seems that the first order Markov process is a good model for the luminance DC coefficients of test image B "Peppers".

"*Peppers*" Chrominance DC Coefficient: The DC coefficients' distribution for the  $C_b$  chrominance value has a mean of -13.94 and a standard deviation of about 5.13 and that for the  $C_r$  chrominance value has a mean of 9.24 and a standard deviation of 14.66.

The "*Peppers*" test image has two main colours, red and green and this will be reflected in the values of the chrominance coefficients. A positive value in the chrominance component  $C_r$  will indicate the presence of the colour red and the presence of the colour green will be reflected in both the  $C_r$  and  $C_b$  chrominance components as negative values (from the RGB to  $YC_bC_r$  conversion matrix).

Figure 7.24 shows a typical line of the DC coefficients for the  $C_r$  chrominance components. The red colour is very strong in the middle of the line which can be expected from the position of the line in the image "*Peppers*" (along the main red capsicum). Figure 7.26b and 7.26c are the correlograms of the  $C_r$  and  $C_b$  chrominance DC components respectively. Since the chrominance coefficients have been down-sampled as compared to the luminance values, the physical separation is further increased and the one-step correlation would not be as high as that for the luminance component.




"Peppers" AC Transform Coefficients: The AC coefficients studied are the coefficient F(0,1), coefficient F(1,0) and coefficient F(1,1). A typical line of the AC coefficient F(0,1) is as shown in Figure 7.25. The values seems to be more random and a lower correlation, if any, is expected.



"Peppers" AC Coefficient F(0,1): Statistics of the luminance and both the chrominance AC Coefficient F(0,1) are as shown in figure 7.27 at the end of the chapter.

It was argued earlier that because of the form of the basis image for the AC coefficient F(0,1) that correlation would be expected in the vertical direction and it can be seen, as anticipated, that there is significant correlation only in the vertical direction.

"Peppers" AC Coefficient F(1,0): Statistics of the luminance and both the C<sub>r</sub> and C<sub>b</sub> chrominance AC coefficient (1,0) are as shown in figure 7.28.

The basis image for the AC coefficient F(1,0) is shown in figure 7.13. It has been argued earlier that because of the form of the basis image for this coefficient that correlation can only be expected in the horizontal direction and from the correlograms, it can be seen that for this image, there is significant correlation in the horizontal direction but as expected, there is very little correlation in the vertical or diagonal direction.

"Peppers" AC Coefficient F(1,1): Statistics of the luminance and both the chrominance AC coefficient F(1,1) are as shown in figure 7.29.

From the correlograms, it can be seen that for this image, there is no significant correlation in any direction.







#### 7.4.3 Correlation in Test Image C:"Toys".

This test image is much more colourful. As in image "Lena", only the first four coefficients have been considered in this study.

#### DC Coefficients.

"Toys" Luminance DC Coefficient: A typical line of the luminance DC coefficients is as shown in Figure 7.30. The DC coefficients' distribution has a mean of -23.07 and a standard deviation of about 25.6.

Figure 7.34a is the correlogram of the luminance DC coefficients and as with the "*Lena*" image it can be seen that the correlation is good in all directions, horizontally, vertically and diagonally.

For a lag of one the correlation between adjacent luminance DC coefficients vertically is around 0.90. The horizontally separated luminance DC coefficients have a 1-step correlation of 0.88 and finally the diagonally separated luminance DC coefficients also have a 1-step coefficient of 0.81.

The correlation is greater in the horizontal direction which is thought to be due to fact that the image contains quite a lot of horizontal structure. Diagonal





correlation, as expected, is less than that along either the horizontal or vertical direction because of the larger physical block separation.

For both the horizontal and vertical direction, a first order Markov approximation for the correlation coefficient of 0.85 is quite a good fit. In the diagonal direction, it can be modelled by a correlation coefficient of 0.7.

Figure 7.31 compares the correlation function with the first order Markov process and it seems to be a good model for the luminance DC coefficients of test image C "*Toys*".

"Toys" Chrominance DC Coefficient: The DC coefficients' distribution for the  $C_b$  chrominance value has a mean of -2.25 and a standard deviation of about 8.53 and that for the  $C_r$  chrominance value has a mean of -1.13 and a standard deviation of 9.3.

The "*Toys*" test image has a lot of very strong colours and the values of the chrominance DC coefficient should be quite similar so that a high inter-element correlation can be expected.

Figure 7.32 shows a typical line of chrominance DC coefficients. As expected, the difference between successive chrominance DC coefficients are quite small.



Figure 7.34b and 7.34c are the correlograms of the  $C_r$  and  $C_b$  chrominance DC components respectively. As previously stated the chrominance DC coefficients have a physical separation of 16 pixels and it would not be expected that the one step correlation would be as high as that for the luminance component. In fact the measured correlation is quite high and compares well with that for the luminance DC coefficients.

#### AC Coefficients.

"Toys" AC Transform Coefficients: The AC coefficients studied are the coefficient F(0,1), coefficient F(1,0) and coefficient F(1,1). A typical line of the AC coefficient F(0,1) is shown in Figure 7.33. The values seems to be more random and a lower correlation, if any, is expected.



It was argued earlier that because of the form of the basis image for the AC coefficient F(0.1) that correlation would be expected only in the vertical direction and it can be seen that this is so.



"Toys" AC Coefficient F(1,0): Statistics of the luminance and both the C<sub>r</sub> and C<sub>b</sub> chrominance AC coefficient (1,0) are as shown in figure 7.36.

The basis image for the AC coefficient F(1,0) is shown in figure 7.13. It has been argued earlier that because of the form of the basis image for this coefficient that correlation can only be expected in the horizontal direction and from the correlograms it can be seen that for this image there is significant correlation in the horizontal direction but as anticipated, there is very little correlation in the vertical or diagonal directions.

"Toys" AC Coefficient F(1,1): Statistics of the luminance and both the chrominance AC coefficient F(1,1) are as shown in figure 7.37.

From the correlograms, it can be seen that for this image there is no significant correlation in any direction.

#### 7.5 SUMMARY OF THE STATISTICAL MEASUREMENTS.

#### 7.5.1 Conclusion on Modelling of Distributions.

Some conclusions can be drawn regarding modelling of AC coefficients in natural images. Of the two distributions studied, the two-sided Laplacian fits the observed data very well, with the worst case having a correlation coefficient of 0.74 and the best correlation coefficient of 0.90. This compares with the Gaussian which generated a worst case correlation coefficient of 0.51 and a best correlation coefficient of 0.79.







If the exercise were to generate a stream of DCT compressed data, modelling the stream of AC coefficients as a Laplacian distribution would be sensible and appropriate.

For the understanding of the AC coefficients, modelling of the distributions as a Gaussian would be beneficial as the Gaussian distribution is mathematically well understood. The range of observed data in the middle fits both a Laplacian and Gaussian distribution very well. The two main areas where the observed data deviates from a Gaussian model are;

- The peakiness of the AC distributions around zero, partially due to the quantisation of the coefficients, which results in a stronger variation from a Gaussian model.
- The long tail-off of the distribution which would result in a lower correlation coefficient when compared to a Gaussian model.

#### 7.5.2 1-Dimensional Correlation of DCT Images.

The DC coefficients are seen to be highly dependent upon the profile of the distribution. The shape of the pixel distribution is carried over to the distribution of the DC coefficients. Therefore, significant correlation can be expected from surrounding coefficients and this is shown to be so.

For the AC coefficients, the first-order pdf tends towards a Gaussian distribution. This implies that the AC coefficients are independent random variables, which could be worrying. If the AC coefficients are independent, it would be impossible to predict any coefficient from knowledge of the surrounding AC coefficients. However, if the correlation exists only in one direction but not in any other, the two-dimensional histogram of a three dimensional image array would not be able to reflect the dependence for that particular direction. The results for the first few coefficients shows that correlation does exist only in a specific direction for a certain AC coefficient.

#### • DC coefficients

For the DC coefficients of the Discrete Cosine Transform, the correlation between successive blocks is very good in the horizontal, vertical and diagonal direction.

In the spacial domain if it is considered that a suitable model to represent the discrete pixel values is a first order Markov Process then since the DCT is a

linear transformation, it seems reasonable to model the spectral coefficients by the same process and in fact the measured results confirm this. The coefficients of the DCT are linearly weighted sums of the pixel intensities and hence because of the Central Limit Theorem it might be expected that the pdf of these should tend towards a Gaussian distribution. However this is not necessarily so because of the strong correlation between adjacent pixels.

#### • AC Coefficients

As expected, the correlation coefficients for successive AC coefficients F(0,1) in the vertical direction is significant but only in the vertical direction whereas the correlation coefficients for successive AC coefficients F(1,0) in the horizontal direction is significant but only in the horizontal direction.

From the shape of the basis image for the AC coefficient F(1,1) no significant correlation is expected in any direction and none of the images showed any significant correlation.

# CHAPTER 8

# CHAPTER 8 : MODIFICATION AND IMPLEMENTATION OF AN ERROR CONCEALMENT ALGORITHM.

In chapter 6, a method using linear interpolation of neighbouring blocks provided the best results for the concealment of video cell loss. Analysing the results shows that most of the replacement blocks are smooth, without very strong edges or orientation. This implies that these contain low frequency detail but limited high frequency components. If a given coefficient is significant only in one of the neighbouring blocks, linear interpolation will have the tendency to average out this value. Therefore it is hardly surprising that the values of the "high frequency" transform coefficients in the replacement blocks are small or zero. The following sections will try to simplify the error concealment algorithm.

# 8.1 MODIFICATION TO THE ERROR CONCEALMENT ALGORITHM.

From statistical studies of some natural images in the previous chapter, the distribution of each transform coefficient in an image is shown to be less correlated as it moves away from the DC transform coefficient F(0,0) towards higher order AC values. Most of the transform coefficients from F(4,4) through to F(7,7) tend to be zero partly due to the large quantisation factors but mainly due to the lack of "high frequency" detail in natural images. The quantisation factors for the transform coefficients are based on the human perception system, and "higher frequency" information which human eyes are less sensitive too are reduced proportionally.

In natural images, most "high frequency" detail tends to cluster within areas of one or a few neighbouring blocks. Looking at a contour map of that transform



coefficient will reveal a flat surface except for small areas where the coefficient is significant.

Distributions of either the DC or first few transform coefficients have been shown in the previous chapter to have significant values for most areas within an image. Surfaces of the contour maps vary smoothly and the correlation between neighbouring transform coefficients are significant.

The initial method of bilinear interpolation uses information from the four nearest neighbouring blocks and this is shown in figure 8.1 for the luminance components. Each transform coefficient is calculated from equation (8.1) and the equation is used to calculate an estimate for all 64 lost luminance transform coefficients.

Luminance,  

$$F_{L1}(i,j) = \frac{2}{7}F_{t1}(i,j) + \frac{2}{7}F_{b1}(i,j) + \frac{2}{7}F_{l2}(i,j) + \frac{1}{7}F_{r1}(i,j) \qquad \dots (8.1)$$

$$F_{L2}(i,j) = \frac{2}{7}F_{t2}(i,j) + \frac{2}{7}F_{b2}(i,j) + \frac{1}{7}F_{l2}(i,j) + \frac{2}{7}F_{r1}(i,j)$$
where  $i, j = 0.1...7$ 

Chrominance,  

$$F_{L}(i,j) = \frac{1}{4}F_{t}(i,j) + \frac{1}{4}F_{b}(i,j) + \frac{1}{4}F_{l}(i,j) + \frac{1}{4}F_{r}(i,j) \qquad \dots (8.2)$$
where  $i, j = 0, 1, \dots, 7$ 

The chrominance values are sub-sampled in the horizontal direction, therefore there is only one  $C_r$  and one  $C_b$  chrominance block in each macroblock. A macroblock is specified as 16  $\times$  8 pixels and consists of two luminance blocks, one  $C_r$  and one C<sub>b</sub> chrominance block. As shown in figure 8.2 the top, left and bottom, right neighbouring blocks are used to estimate the lost block in the middle. The equation (8.2)is used for estimating each of 64 chrominance the transform coefficients.

The algorithm is modified by reducing the number of



coefficients used for estimating the lost block to the DC Transform coefficient and the first three AC transform coefficients F(0,1), F(1,0) and F(1,1). A study of the correlation between the same transform coefficient in adjacent blocks was described in section 7.4 of the previous chapter. The results register significant correlation in specific direction for the first few transform coefficients and only for the first few coefficients.

#### Luminance,

$$\begin{split} F_{L1}(0,0) &= \frac{2}{7} F_{t1}(0,0) + \frac{2}{7} F_{b1}(0,0) + \frac{2}{7} F_{l2}(0,0) + \frac{1}{7} F_{r1}(0,0) \qquad \dots (8.3) \\ F_{L1}(1,1) &= \frac{2}{7} F_{t1}(1,1) + \frac{2}{7} F_{b1}(1,1) + \frac{2}{7} F_{l2}(1,1) + \frac{1}{7} F_{r1}(1,1) \\ F_{L1}(0,1) &= \frac{1}{2} F_{t1}(0,1) + \frac{1}{2} F_{b1}(0,1) \\ F_{L1}(1,0) &= \frac{2}{3} F_{l2}(1,0) + \frac{1}{3} F_{r1}(1,0) \end{split}$$

 $F_{L2}(0,0) = \frac{2}{7}F_{l2}(0,0) + \frac{2}{7}F_{b2}(0,0) + \frac{1}{7}F_{l2}(0,0) + \frac{2}{7}F_{r1}(0,0)$   $F_{L2}(1,1) = \frac{2}{7}F_{l2}(1,1) + \frac{2}{7}F_{b2}(1,1) + \frac{1}{7}F_{l2}(1,1) + \frac{2}{7}F_{r1}(1,1)$   $F_{L2}(0,1) = \frac{1}{2}F_{l2}(0,1) + \frac{1}{2}F_{b2}(0,1)$   $F_{l2}(1,0) = \frac{1}{3}F_{l2}(1,0) + \frac{2}{3}F_{r1}(1,0)$ 

Chrominance,

$$F_{L}(0,0) = \frac{1}{4}F_{t}(0,0) + \frac{1}{4}F_{b}(0,0) + \frac{1}{4}F_{l}(0,0) + \frac{1}{4}F_{r}(0,0) \qquad \dots (8.4)$$

$$F_{L}(1,1) = \frac{1}{4}F_{t}(1,1) + \frac{1}{4}F_{b}(1,1) + \frac{1}{4}F_{l}(1,1) + \frac{1}{4}F_{r}(1,1)$$

$$F_{L}(0,1) = \frac{1}{2}F_{t}(0,1) + \frac{1}{2}F_{b}(0,1)$$

$$F_{L}(1,0) = \frac{1}{2}F_{l}(1,0) + \frac{1}{2}F_{r}(1,0)$$

Only the first four transform coefficients for both luminance and chrominance blocks are used to generate the replacement for the lost block. The luminance and chrominance blocks are calculated from equation (8.3) and (8.4) respectively.

#### 8.2 RESULTS FROM THE SIMULATION.

#### 8.2.1 Comparsion of Methods.

Colour Plate 8.1 consists of two sets of images taken from *image sequence 2* "Shoes" used in chapter 6. In each image 2% of the total number of macroblocks are corrupted. The first image in each set has errors replaced by black macroblocks. The second image in each set replaces errors by information derived from the neighbouring blocks. Images replaced with the modified algorithm are quite acceptable. On close examination, certain individual errors are noticeable. But on the whole, the majority of the errors are well hidden. The next section contains magnified portions of images and compares the earlier algorithm with the modified version.

#### 8.2.2 Comparsion of Methods (Magnified).

Colour plate 8.2 is a comparison of images generated by the three methods. Method 1 is an image with errors replaced by black macroblocks. Method 2 replaces each lost macroblock with linearly interpolated data from the neighbouring blocks. Method 3 is similar to method 2 but only a subset of the transform coefficients are used for generating the information required in the replacement macroblock.

Four sets of images are shown here with three images in each set. Each image zooms in on an area  $\frac{1}{25}$  of the original image and the image is then magnified to fit the space allocated in colour plate 8.2. The images are taken from the image sequence 2 "SHOES" used in chapter 6.

The first set of images show a brand name "PEPE" with two errors. Each macroblock error covers a substantial area of a character. This is a good example of "high frequency" detail which is localised within each block. Bilinear



Colour Plate 8.1

# IMAGE SEQUENCE 2: "SHOES"

÷.

Comparison between Error Concealment Methods



Colour Plate 8.2

interpolation of the neighbouring blocks will not generate the detail which might have been available in the lost block. It will not be possible to replace the error with a close enough replica unless additional information is transmitted from the transmitter. The replacement blocks generated for both methods 2 and 3 are, however, subjectively acceptable substitutes when viewed at a normal viewing distance. The replacement is noticeable but visually acceptable.

The second set of images zooms in on a part of the shoe with alternate blue and white strips. The strips are at an angle to the horizontal and method 2 using all the 64 transform coefficients in the neighbouring blocks produces a replacement block that fits in well with the surrounding. The replacement block generated by method 3 uses only four transform coefficients and the results are not as good as in method 2.

For the third set of images, method 1 shows the errors replaced by black blocks. Both method 2 and 3 generated satisfactory replacement blocks for the lost macroblocks in the image. For areas where surrounding blocks are similar, a close replica can be generated by the method of linear interpolation. There are hardly any errors that are visually noticeable.

The last set of images shows errors in the green banister-like structure. Methods 2 and 3 replace the lost macroblocks with information generated by linearly interpolated data from the neighbouring blocks. These replacement blocks are a good substitute for the lost blocks. Only one of the replacement blocks is noticeable, this is where a part of the white background is carried forward to the green structure.

#### 8.2.3 Conclusions and Discussions on the Modified Algorithm.

If a particular transform coefficient in one or two of the neighbouring blocks is significant and the value of the same transform coefficient in the other neighbouring blocks is small in comparison, linear interpolation of the blocks will reduce the significant value by a factor of at least one-half. This is a likely scenario for uncorrelated or weakly correlated sequences. "High frequency" transform coefficients within the image are usually weakly correlated or uncorrelated. Therefore, most of the linearly interpolated "high frequency" transform coefficients for the replacement block will be zero.

Statistical studies show correlation exists for the DC and first few AC transform coefficients. Linearly interpolating lost data from these coefficients is probably sensible. Therefore the modified error concealment algorithm concentrates on just the DC coefficient F(0,0) and AC coefficients F(0,1), F(1,0) and F(1,1).

Results obtained seem to confirm the above. Using just the DC and first three AC transform coefficients generates results almost as good as the original algorithm. As shown in colour plate 8.1, most of the images generated by method 3 are comparable with images generated by method 2, the original algorithm. The obvious difference is in the second set of images, where the replacement block for the original algorithm contains significant "high frequency" coefficients. This is because the surrounding blocks are similar in structure and the "structure" will be carried over to the replacement block. The modified algorithm in method 3 uses only a few coefficients and the "structure" cannot be carried by just the "low frequency" coefficients. The resulting replacement block is of an inferior representation as compared to the replacement block of method 2.

The main conclusion is that the majority of the replacement blocks generated by the modified error concealment algorithm are satisfactory. Using a subset of the transform coefficients is adequate and a reduction in the computational complexity is achieved.

#### 8.3 IMPLEMENTATION OF THE CONCEALMENT ALGORITHM ON THE MASTER PROJECT.

The MASTER project, which is described in appendix C [43,44], uses M-JPEG compression hardware which was designed primarily for used on a standalone machine, where images are transfered to and from the hard-disk. This is a reliable medium and the bit error rate is very low. The compression cards need substantial modifications to operate in a network environment and the resilience to cell loss must be improved. For this compression hardware, an undetected error in a frame would result in the lost of all subsequent information in the frame. At times, the loss may be unrecoverable and the whole system would collapse.

An algorithm was implemented on the MASTER project to determine how long it would takes to run the concealment technique as a process on a transputer. Minimal changes to the system were allowed and the error concealment algorithm was applied as a process in the receiver. As shown in figure 8.3, the cells arriving from the MATMX (Model Asynchronous Transfer Mode eXchange) [45] are sent to the Reassembly process. After reassembly, the error concealment algorithm is applied to conceal the errors, if errors are detected in the cell stream. After concealment, a valid image file is available for the JPEG joining and MediaSpace control processes.

Nothing is impelemented to aid error concealment at the transmitter and no additional side information is added to the video cell stream. Reset markers, generated by a modulo 8 counter in the JPEG file format were activated and used for separating minimum code units (MCUs) in



the image. This ability to locate individual minimum coding unit within an image is useful and necessary both for the error concealment algorithm and JPEG joining of two or four images to form a larger image. A full implementation would include a block interleaving technique to reduce the effect of consecutive losses.

An occam program was written to implement the error concealment algorithm as a single process on a T805 transputer. This also included a section in the reassembly layer which checks for errors.

A complete frame in the image sequence fits into an AAL5 (ATM Adaptation Layer) packet. At the segmentation layer, data in each cell carries a portion of the AAL5 packet. At the end of the AAL5 packet, an additional "end of packet" cell is transmitted which is identified by a special header bit being set. This cell contains the length of the packet in the data field. The AAL5 checksum was not implemented.

At the receiver, the reassembly process will collect the correctly received ATM cells, strip off the header and reassemble the data to form the AAL5 video packet. The last "end of packet" cell will contain the length of the packet and this value is compared with the physical length of the AAL5 video packet. Any discrepancy will indicate an error in the received video packet. The whole packet will then be sent to the error concealment subroutine.

The error concealment algorithm commences by searching through the data stream for reset markers, which are identified by FFD0 (Hex) - FFD8 (Hex) and provide a modulo 8 count. A break in the count will indicate the loss of one or

more consecutive macroblocks. A lost of n macroblocks, where n is greater than 8, would be indistinguishable from a lost of n modulo 8 macroblocks.

In the MASTER Project the image size is  $192 \times 144$  pixels and each image is divided into 216 macroblocks for coding by the JPEG compression algorithm. The number of B-channels used for transmitting this information over primary rate ISDN is set at ten, corresponding to a bandwidth of 640 Kbytes/sec. This works out to an average of approximately 50 - 70 ATM cells allocated for each image. The actual number of cells is dependent upon the image content and the quantisation factor used during the image compression. In this implementation, there is no block interleaving at the transmitter and the loss of each cell would equate to about three to five macroblocks.



After detecting the errors, the program can set about concealing them. The position of each lost macroblock is recorded together with information from the macroblocks directly above and below the lost macroblock. For this simplified implementation, only the top and bottom macroblock is used for the error concealment algorithm. As consecutive macroblocks are lost, it would not be beneficial to look for correctly received left and right macroblocks.

The entropy code segment in each correctly received macroblock is sent to a subroutine which contains the look-up tables for both the Variable Length Codes and Huffman Codes. By looking at the relevant columns, decoded information can be extracted and used to generate the DCT transformed blocks. There is no need to do an inverse DCT as the error concealment algorithm is performed in the transform domain.

Information in the transform domain for both the top and bottom blocks are used to generate the replacement block. The two luminance and one  $C_r$  and  $C_b$ chrominance lost blocks are replaced with linearly interpolated data from the corresponding top and bottom received blocks. Only the DC coefficient F(0,0)and first three AC coefficients F(0,1), F(1,0) and F(1,1) are used in the algorithm.

After calculating the replacement information for the block, this information needs to be encoded into a series of Huffman and Variable Length Codes. A subroutine reads the coefficients in the transformed block and searches the appropriate look-up table for the Huffman or Variable Length Codes. The two luminance and one  $C_r$  and  $C_b$  chrominance blocks are then encoded in the correct order and the regenerated entropy coded segment can be used as a substitute for the lost macroblock.

Each of the lost macroblocks is replaced by the same method and the valid frame is sent to the next transputer process in the pipeline.

#### 8.4 RESULTS OF THE IMPLEMENTATION.

For this section of the thesis, the main interest is in testing the algorithm on a working videoconferencing system to see if the algorithm can be implemented in real time. Real time in this case implies processing at a rate faster than full motion video. At a rate of 25 frames/sec, the error concealment algorithm must be executed in less than 40 ms.

The loss of a cell in every frame corresponds to a cell lost rate of about 10<sup>-2</sup>. A lost cell is simulated by dropping a cell in the AAL5 data packet. In the

MASTER project videoconferencing images were compressed by a factor of 25:1-30:1, therefore in each cell there are about three to five macroblocks.

In the error concealment subroutine, the internal clock count within the transputer was recorded at the start and end of a 10 seconds interval. The difference between the two recorded counts can be converted into time and this is divided by 250 to give the time required to correct for errors in a single frame. This works out to an average of about 26 milliseconds for a single frame.

For each macroblock, about 6 milliseconds is required to generate a replacement. This includes the time to locate the error, perform entropy decoding of two neighbouring blocks, linear interpolation of the neighbouring block and finally performed entropy encoding of the generated replacement block.

The error concealment algorithm works in real time and can be used in this implementation to correct up to approximately 6 macroblocks in each frame when implemented as a single process in a T805 transputer. To conceal for more errors, it would be necessary to acquire more computational power by inceasing the number of transputers and introducing parallellism into the algorithm.

# CHAPTER 9 ▷▷▷



## CHAPTER 9 : THE IMPORTANCE OF SPECIFIC TRANSFORM COEFFICIENTS.

Sections of chapter 6 and chapter 8 described the initial and modified method of error concealment using information derived from neighbouring blocks. An image with errors could not be displayed without some form of error correction or concealment. Linear interpolation of the four neighbouring blocks, namely top, bottom, left and right, provided the best method of estimation for the missing block. Two areas where this method of concealment do not work well are;

- When the lost block contains high frequency details which are localised and uncorrelated with neighbouring blocks. Error concealment would not be able to generate the high frequency details without the transmission of additional side information.
- When the lost block cuts across an edge boundary, the replacement block is usually not very effective. There are contrasting pixels on both sides of the edge and linear interpolation tends to average out the values in the neighbouring blocks. The replacement block breaks the continuous edge and it can be visually distracting.

Concealment methods will not be able to correct for the first type of error. To improve upon the algorithm, the work concentrated on reducing concealment errors of the second type. At the receiver, concealment errors of the second type can be improved with additional local information derived from the neighbouring blocks.

This chapter will look into the significance of selected transform coefficients and how they influence the content within an image block. Some initial work to identify simple edges in a transformed block has been suggested and will be reported in the next section. The second section describes subjective and objective measurements of images with different numbers of transform coefficients.

#### 9.1 CLASSIFICATION OF BLOCKS INTO SIMPLE SHAPES.

In this section, the main objective is to classify the transformed blocks within an image into various simple shapes. Classification will be based on the strength of selected transform coefficients and this section will identify the crucial transform coefficients common to blocks with strong edges.

#### 9.1.1 Four Basic Classes.

Blocks are allocated to different classes and what is of interest is how the various shapes in the pixel domain influence the energy distributions within the transform domain. The block sizes are defined as  $8 \times 8$  and the initial work concentrate on shapes but not texture. Classification is based only on the luminance values in a block and four classes are defined as flat, horizontal, vertical or diagonal.



#### Class 1: Flat Blocks.

A block where the pixels vary within a small range of intensities is classified as a flat block. Visually, the block would appear to have the same shade with no prominent edges across the block. In the transform domain, only the DC coefficient F(0,0) would be significant, the AC coefficients would have small and insignificant values.

#### Class 2: Vertical Blocks.

A vertical block would appear to have two contrasting vertical strips. In the pixel domain, pixels along the y-axis would have similar intensities but pixels along the x-axis will change substantially at the edge boundary.

An example of a vertical block with a very strong straight edge along the middle of the block is shown below together with the cosine transform coefficients before and after quantisation. A Hadamard transform would compact the information of this particular pixel block into just one AC coefficient F(0,1). However, a cosine transform would decompose the information within a pixel block into a series of cosinusoidal waveforms. Therefore, the information is spread among the transform coefficients F(0,1), F(0,2)...F(0,7) and the rest of the coefficients would be zero. As the block exhibits vertical anti-symmetry, it can be shown mathematically that the transform coefficients of even rows would be zero.

Pixel Domain	Transform Domain	After Quantisation
128         128         128         -128         -128         -128         -128           128         128         128         -128         -128         -128         -128           128         128         128         128         -128         -128         -128           128         128         128         128         -128         -128         -128           128         128         128         128         -128         -128         -128         -128           128         128         128         128         -128         -128         -128         -128           128         128         128         128         -128         -128         -128         -128           128         128         128         128         -128         -128         -128         -128           128         128         128         128         -128         -128         -128           128         128         128         128         -128         -128         -128           128         128         128         128         -128         -128         -128	→	$\Rightarrow \begin{array}{cccccccccccccccccccccccccccccccccccc$
128 128 128 128 -128 -128 -128 -128	Lo o o o o o o _	_

Due to the Mach Band effect most visually strong edges in natural images will still have a gradual change of pixel values along the edge boundary and not the abrupt change shown in the above example. Nevertheless, most distinct vertical blocks can be predicted from strong components in AC coefficients F(0,1) and F(0,2), mainly F(0,1).

#### Class 3: Horizontal Blocks.

Examples of horizontal blocks are shown in figure 9.1. In the pixel domain, pixels along the x-axis would have similar intensities but pixels along the y-axis will change substantially at the edge boundary.

		Pix	el D	Oom	ain				Tran.	sfi	or	m	L	)0	m	aiı	n A	After	Q	)u	ar	ıti	sa	tì	on
-128 -128 -128 -128 128 128 128 128 128	-128 -128 -128 -128 128 128 128 128 128	-128 -128 -128 -128 128 128 128 128 128	-128 -128 -128 -128 128 128 128 128 128	-128 -128 -128 128 128 128 128 128	128 128 128 128 128 128 128 128 128	-128 -128 -128 -128 128 128 128 128 128	-128 -128 -128 -128 128 128 128 128 128	-	0 928 0 -326 0 218 0 -185	0 0 0 0 0 0 0 0	000000000	000000000000000000000000000000000000000	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	000000000000000000000000000000000000000	00000000	⇒	0 84 0 -20 0 5 0 -3	000000000000000000000000000000000000000	0 0 0 0 0 0 0 0	000000000	0 0 0 0 0 0 0 0 0	000000000000000000000000000000000000000	0 0 0 0 0 0 0	000000000000000000000000000000000000000

Similarly, prominent horizontal blocks would have strong components in AC coefficients F(1,0) and F(2,0). It would be sensible to concentrate on just the AC coefficient F(1,0) when checking a block for horizontal structure.



#### Class 4: Diagonal Blocks.

The four main types of diagonal blocks are shown in figure 9.1. The first two have edges along the main diagonal and the last two have off diagonal edges.

		Pi	xel I	Don	nain	ţ					Tra	nsfo	rm	Do	та	in			Aft	er g	Qu	an	tis	at	io	n
0 128 128 128 128 128 128	-128 0 128 128 128 128 128 128	-128 -128 0 128 128 128 128	-128 -128 -128 0 128 128 128	-128 -128 -128 -128 0 128 128	-128 -128 -128 -128 -128 0 128	-128 ~128 -128 -128 -128 -128 -128 0	-128 -128 -128 -128 -128 -128 -128 -128		-	0 -583 0 -61 0 -18 0	583 0 268 0 48 0 15	0 268 0 -152 0 -30 0	61 0 152 0 -98 0 -19	0 48 0 98 0 -66 0	18 0 30 0 66 0 -42	0 15 0 19 0 42 0	5 0 7 0 10 0 22	-	0 -49 0 -4 0 -1 0	53 0 21 0 2 0 0	0 19 0 -7 0 -1 0	4 6 0 -2 0 0	0 2 0 2 0 -1 0	1 0 1 0 1 0	0 0 0 0 0 0 0	000000000000000000000000000000000000000
L 128	128	128	128	128	128	128	0_	ł		L -5	0	-7	0	-10	0	-22	۰_		Ľ٥	0	0	0	0	0	0	٦0

An example is shown here for a pixel block with a main diagonal edge. The transformed matrix has significant values along the line of entries lying immediately above and below the main diagonal. The four coefficients with the largest values are F(1,0), F(0,1), F(2,1) and F(1,2). The transform coefficients along the main diagonal would not be zero if pixels along the main diagonal are non-zero or pixels on opposite sides of the diagonal have contrasting but different absolute values. Significant values in coefficients F(0,0), F(1,1) and the other four F(1,0), F(0,1), F(2,1) and F(1,2) would be sufficient to detect a diagonal structure.

		Pix	cel I	Dom	ain				Tra	nsfc	rm	Da	ome	in			Af	ter	Qu	ar	itis	a	tic	m
28 1 28 1 28 1 28 1 28 1 28 1 28 1 28 1	128 128 128 128 128 128 128 0 -128	128 128 128 128 128 128 0 -128 -128	128 128 128 128 0 -128 -128 -128	128 128 128 0 128 128 128 128	128 128 0 128 128 128 128 128	128 0 -128 -128 -128 -128 -128 -128 -128	0 -128 -128 -128 -128 -128 -128 -128 -128	0 583 0 61 0 18 0 5	583 0 268 0 48 0 15 0	0 268 0 152 0 30 0 7	61 0 152 0 -98 0 -19 0	0 -48 0-98 0 66 0 10	18 0 30 66 0 -42 0	0 -15 0 -19 0 -42 0 22	5 0 7 0 10 0 22 0	-	0 49 0 4 0 1 0	53 0 -21 0 -2 0 0 0	0 -19 0 7 0 1 0	4060 <u>-</u> 000	0 -2 0 -2 0 1 0 0	1 0 1 0 0 0	0 0 0 0 0 0 0 0	00000000

Similar results are shown for pixel blocks with off diagonal edges. The same six transform coefficients F(0,0), F(0,1), F(1,0), F(1,1), F(2,1) and F(1,2) would be used to identify a diagonal structure.

#### 9.2 IMAGE IMPAIRMENT DUE TO THE LOSS OF SELECTED TRANSFORM COEFFICIENTS.

This section attempts to establish the significance of certain transform coefficients to the quality of an image. Objective and subjective measurements were taken for images with different selected transform coefficients.



Experiments on the human visual response to sine-wave and square-wave spatial distributions, viewed at a range of spatial frequencies, were conducted as early as the 60s and the results indicated the importance of low spatial frequencies in the human visual contrast phenomena [46,47]. A Human Visual System (HVS) model can be derived by taking the inverse of the contrast sensitivity curve obtained from psychovisual experiments.

Figure 9.1 is a plot of the HVS model and various studies have since been conducted on the use of the HVS model in the encoding of images and modifications to the visual model for cosine transform images have also been recorded. [48,49].

The JPEG algorithm as specified in the document DIS 10918 "Digital Compression and Coding of Continuous-tone Still Images" [2] recommended two

quantisation tables for use in JPEG compressed images. The tables are derived with the aid of the HVS model and are shown here in figure 9.2. Quantisation in the luminance table emphasis values to the left of the off-diagonal particularly the DC coefficient F(0,0) and the nearest few "low" frequency" AC coefficients. The lower values allocated to the first row as compared to values in the first column implies that a higher amount of vertical structure is expected from most images. For the chrominance quantisation table. the only first 15 coefficients in the zigzag scan are of any significance.







	Luminance Quantisation Table									Chrominance Quantisation Table								
٢								1	ſ									
16	п	10	16	24	-40	51	61		17	18	24	47	99	99	99	99		
12	12	14	19	26	58	60	55		18	21	26	66	99	99	99	99		
14	13	16	24	40	57	69	5 <del>6</del>	1	24	26	56	99	9 <b>9</b>	99	99	99		
14	17	22	29	51	87	80	62		47	66	99	99	99	99	99	99		
18	22	37	56	68	109	10 <b>3</b>	77		99	99	99	99	99	99	99	<del>9</del> 9		
24	35	55	64	81	104	113	92		99	99	99	9 <b>9</b>	99	99	99	99		
49	64	78	87	103	121	120	101		99	99	99	99	99	99	99	99		
72	92	95	98	112	100	103	99		99	<b>99</b>	9 <b>9</b>	99	99	99	9 <b>9</b>	99		
ς								7	ι.									

#### 9.2.2 Estimation of Image Quality.

It is obvious that the DC and "low frequency" AC coefficients are of greater importance and it would be interesting to see how the withdrawal of certain transform coefficients affect the image quality.

Image quality is of course, very much a subjective, as well as an objective measure, and needs to be expressed quantitatively as well as qualitatively. The subjective measure used in this experiment is that of the impairment scale, which is explained in the next section. For the objective measures, the Mean Square Error (MSE) and Normalised Mean Square Error (NMSE) are used when assessing the images.

#### 9.2.3 Test Images.

A total of 3 test images were selected for the assessment. The three images used were *Image 1:"Lena"*, *Image 2:"Mandrill"* and *Image 3:"Peppers"* which are standard images used in various experiments. The images were then compressed using the JPEG (Joint Photographic Expert Group) still picture compression algorithm.

Eight images were then generated from each of the test images, producing a total of 24 still images. For each of the eight images, a different number of transform coefficients were selected for decoding the compressed images. The test images were generated according to the descriptions in table 9.1.

Image.	Description.
(a)	Original.
(b)	Only the DC coefficient F(0,0).
(c)	DC and the first two AC coefficients $F(0,0)$ , $F(0,1)$ & $F(1,0)$ .
(d)	DC and the nearest three AC coefficients $F(0,0)$ , $F(0,1)$ , $F(1,0)$ & $F(1,1)$ .
(e)	DC and 5 selected AC coefficients F(0.0), F(0.1), F(1.0), F(1.1), F(2.1) & F(1.2).
(f)	First 6 coefficients in zigzag scan F(0,0), F(0,1), F(1,0), F(1,1), F(2,0) & F(0,2).
(g)	First 10 transform coefficients in the zigzag scan.
(h)	First 21 transform coefficients in the zigzag scan.
	Table 9.1: Images generated for Subjective Assessment

Colour plate 9.1 shows the three test images and the impairment score sheet and Colour plate 9.2 shows one complete set of images generated for the subjective and objective assessment. Only a portion of the actual "Lena" image is shown. This portion is then magnified and displayed at four times its actual size. This will emphasis the image impairments visible to the eye and most subjects

### IMAGES FOR IMPAIRMENT STUDY

JPEG Compressed Still Images



Colour Plate 9.1



#### Colour Plate 9.2

suggested that this image, containing the human face, is the easiest to identify and to criticise.

#### 9.2.4 Subjective Assessment.

The merit of the impairment scale is that it can be used to appraise the various images, where each image has a particular impairment that has been previously identified and defined. It is useful when it is desired to rate the magnitude of a particular type of impairment in the presence of other types. The scale is divided into five grades, as recommended by CCIR and shown in table 9.2 [50].

Impairment Scale.											
5	Imperceptible.										
4	Perceptible, but not annoying.										
3	Slightly annoying.										
2	Annoying.										
1	Very annoying.										
T	able 9.2: CCIR recommended Impairment Scale										

A subjective assessment was conducted and twenty subjects were involved in the experiment, six were considered to be experts and fourteen as non-expert. An expert is defined as a person who has been involved in television, photographic or visual arts work and is sometimes involved in critical examination of picture.

As much as possible, the picture conditions specified in CCIR recommendation 500 were followed. The subject was helped to "anchor" his quality ratings by telling him that Image (a), the original image, is the best and to be given a quality rating of "5". The subject was presented with the pictures in random order and was asked to rate them according to quality. He was allowed to view each picture at a distance of four to six times image height and for as long as he liked.

Since several subjects were used in the evaluation process, the mean rating is given by:

$$R = \sum_{k=1}^{n} s_{k} n_{k} / \sum_{k=1}^{n} n_{k} \qquad ...(9.1)$$

where  $s_k$  is the score associated with the  $k^{\text{th}}$  rating,  $n_k$  is the number of observers with this rating, and n is the number of grades in the scale.

Table 9.3a - Subjective Results for Image 1: "Lena"															
	Img (1a)	Img (1b)	Img (1c)	Img (1d)	Img (1e)	Img (1f)	Img (1g)	Img (1h)							
Expert	5	1	3	2	4	3	4	5							
Non-Expert	5	1	2	2	3	2	4	4							
5 -impercept	5-imperceptible, 4-perceptible, not annoying, 3-slightly annoying, 2-annoying, 1-very annoying.														
	Table 9.3b - Subjective Results for Image 2: "Mandrill"														
	Img (1a) Img (1b) Img (1c) Img (1d) Img (1e) Img (1f) Img (1g) Img (1h)														
Expert	5	1	3	3	4	3	4	5							
Non-Expert	5	1	3	2	4	3	4	5							
5 -impercept	ible, 4-pero	ceptible, not	annoying,	3 -slightly ar	moying, 2-	annoying,	1 -very anno	ying.							
	Table	9.3c - Su	bjective R	Results for	Image 3:	"Peppers"									
	Img (1a)	Img (1b)	Img (1c)	Img (1d)	Img (1e)	Img (1f)	Img (1g)	Img (1h)							
Expert	5	2	3	3	4	3	4	5							
Non-Expert	5	1	3	2	4	3	4	4							
5 -impercept	ible. 4-per	ceptible, not	annoying,	3 -slightly ar	noying, 2	annoving.	1 -very anno	ving.							



Results of the subjective assessment for Image 1, 2 and 3 are tabulated and shown in Table 9.3a, 9.3b and 9.3c respectively. Figure 9.3 plots the result in the form of a chart. The mean rating value of each image is shown together with the average rating of all three images for each combination of coefficients.

From the results it is seen that image type (b) is subjectively the worst. Image type (c), (d) and (f) produced an impairment scale of around "2" and "3", which are classified as "annoying" and "slightly annoying". Both image type (e) and (g) have an average rating of around "4", which is a much higher rating than most of

the other images. Image type (g) has a higher average of the two, which is hardly surprising, as the number of coefficients used for decoding was much higher. Image type (h) for all three images has the highest average rating of "5", which is considered as "*imperceptible*" in the impairment scale.

The most interesting result is the high average score achieved by image type (e). Both image type (e) and (f) use six transform coefficients in the decoding of the image. Image type (f) uses the first six transform coefficients in the zigzag scan order, namely F(0,0), F(0,1), F(1,0), F(2,0), F(1,1) and F(0,2). Image type (e), however uses the six transform coefficients F(0,0), F(0,1), F(1,0), F(1,1), F(2,1)and F(1,2). It should be noted that the quantisation used on the coefficients does not differ significantly and the possibility of errors introduced due to different quantisation factors are minimal.

All the three images of type (e) generate a subjective rating of "4" which is considered as "*perceptible, but not annoying*". If only a limited number of coefficients are to be used to represent the image, the six specially selected coefficients in image type (e) would be sufficient to give a subjectively acceptable image.

#### 9.2.5 Objective Assessment.

The objective measures of interest are the mean square error (MSE) and the normalised mean square error (NMSE) between the input image and output image.

Suppose the input image consists of an  $N \times M$  array of pixels f(x,y), where x = 0, 1, 2, ..., N-1 and y = 0, 1, 2, ..., M-1. The output image, which is a reconstructed image of lower quality, is g(x,y). The error between an input pixel and the corresponding output image pixel is e(x,y), where e(x,y) = f(x,y) - g(x,y).

The mean square error is defined as;

$$MSE = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \left[ e(x, y) \right]^2 \qquad \dots (9.2)$$

and the normalised mean square error is;

$$NMSE = \frac{\frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [e(x,y)]^2}{\frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [f(x,y)]^2} \dots (9.3)$$

The NMSE is a better measurement for comparison between different images. Each value is simply a measurement of the pixel difference between the two images and the calculated NMSE is converted into a percentage value. 0.0%

12.2%

7.9%

NMSE (%)

	Tabl	e 9.4a - 1	Objective	Results fo	r Image 1	: "Lena"		
	Img (1a)	Img (1b)	Img (1c)	Img (1d)	Img (1e)	Img (1f)	Img (1g)	Img (1h)
MSE	0.0	241.8	109.1	105.8	69.9	96.8	47.0	12.3
NMSE (%)	0.0%	10.9%	4.9%	4.8%	3.1%	4.3%	2.1%	0.5%
	Table	9.4b - O	bjective R	esults for	Image 2: "	Mandrill	Img (2g)	Img (2h)
MSE	0.0	378.1	292.6	287.6	233.8	277.8	203.8	103.1
NMSE (%)	0.0%	25.5%	19.7%	19.4%	15.8%	18.7%	13.7%	7.0%
	Table	9.4c - O	bjective R	esults for	Image 3: '	"Peppers"		
	Img (3a)	Img (3b)	Img (3c)	Img (3d)	Img (3e)	Img (3f)	Img (3g)	Img (3h)
MSE	0.0	342.4	222.2	201.7	120.1	1815	89.2	43.15

7.2%

6.5%

4.3%

4.3%

3,2%



Results of the objective measurements for Image 1, 2 and 3 are presented in table 9.4a, 9.4b and 9.4c respectively. Figure 9.4 is a plot of the objective measurements showing only the normalised mean square error.

Image 2: "*Mandrill*" has the worst results of the three images. From the image it is obvious that details around the cheeks, which cover a large portion of the image, will account for the high percentage of AC energy. For images 1 and 3, the total AC energy (all 63 AC transform coefficients) within the image is less than 15% of the total energy (Both AC and DC energy).

Image type (b) is the worst of all the eight image types used in this experiment. In the other image types, the NMSE values show an expected general downward trend as the number of AC transform coefficients increases.
Image type (c) uses the DC and AC coefficients F(0,1) and F(1,0) a substantial drop in the NMSE when compared with image type (b) verifies the importance of these two AC coefficients. For Image 1 and 3, up to 50% of the total AC energy can be retrieved by using just the F(0,1) and F(1,0) AC coefficients.

Image type (c), (d) and (f) all have quite similar results, they use three, four and six transform coefficients respectively but the improvement in image quality is not substantial. However, image type (e) using a different set of six transform coefficients { F(0,0), F(0,1), F(1,0), F(1,1), F(2,1) and F(1,2) } from image type (f) extracts at least an extra 10% of the total AC energy available.

Image types (g) and (h), using 10 and 21 transform coefficients respectively, are good enough reconstructions for image 1: "*Lena*" and 3: "*Peppers*". Using the first 21 transform coefficients in the zigzag scan generates an NMSE of as low as 0.5% for Image (1h) and up to 7% for image (2h).

# 9.2.6 Conclusions on Image Quality Assessment.

Some significant comments from subjects involved in the experiments for subjective measurements are summerised here;

- Image type (b) uses only the DC coefficients and a large majority of the subjects objected to the blockiness within this image.
- The subjective measurements were expected to reflect the lower NMSE objective measurement of image 2 "*Mandrill*" but as can be seen from the tables, the subjective results are quite similar for all three images of the same type. From talking to some of the subjects, it seems then the criterion for rejecting an image is the blockiness within an image rather that the lack of high detail information. The lack of high frequency details is akin to an out-of-focus image and this is visually more acceptable.
- Both groups of experts and non-experts were more critical when accessing image 1: "Lena", resulting in lower impairment ratings for most image types. Most subjects suggested that image 1 was the easiest to grade, which is also the reason why the subjects were more critical in their assessment as it is an image of a human being, something that is easy to appreciate, recognise and criticise.

This study shows the importance of certain transform coefficients in the estimation of image quality. The six coefficients selected for image (e) were chosen from knowledge gained from the study of transform coefficients. The

coefficients were sufficient for generating simple blocks with a horizontal, vertical or diagonal edge across the middle of the block.

This conclusion reinforced the belief that a limited number of transform coefficients will be sufficient to estimate a lost block when applying an error concealment algorithm, and also for reconstructing a good quality image with a limited number of transform coefficients. The quality of the image is controlled by the selection of coefficients in addition to the actual number of coefficients used in its reconstruction.

#### 9.3 A SIMPLIFIED CLASSIFICATION.

The results from section 9.1 confirmed the importance of specific transform coefficients in the generation of flat, horizontal, vertical and diagonal blocks. What is required now is a simplified classification algorithm which checks a minimal number of transform coefficients in order to identify the presence or otherwise of a specific feature in the block. Results from previous sections on objective measurements and block classification indicate the importance of the AC coefficients F(0,1) and F(1,0). As much as 50% of the total AC energy could be concentrated in these two transform coefficients. A simplified classification method is proposed which concentrates on these two "low frequency" AC coefficients only. A strong component in AC coefficient F(0,1) (when compared with coefficient F(1,0)) would indicate a vertical edge across the block. Likewise, a strong component in AC coefficient F(1,0) would be a good indication of a

horizontal block. If both coefficients are greater than a predefined value and the two coefficients are similar, a diagonal block would be expected. If the block fails the three above tests and both coefficients F(0,1) and F(1,0) are below another predefined value the block would be classified as a flat block. This is a rough estimation and some results are shown here.



Sections of the image with strong diagonal, horizontal or vertical edges, such as images A and B in figure 9.6, are easy to classify. Diagonal, horizontal and



vertical blocks along the edge are detected and are shown with their initial. Flat blocks are shown without any changes. For images C and D, some of the edge blocks are not detected. The difference in value between pixels on both side of the edge is not huge. The predefined threshold will miss certain edge blocks and at other times, wrongly identify a visually non-existing edge block. The simple classification needs further investigation.

Other possible mechanisms may calculate the total AC energy within each block and use a ratio of total AC energy as the threshold rather than using a predefined threshold. It might also be sensible to reduce the number of classes and simply identify the blocks as either flat or non-flat blocks.

### 9.4 CONCLUSIONS.

This chapter describes some subjective and objective measurements on image impairment due to the omission of selected transform coefficients. If only a limited number of transform coefficients is to be used to represent the image, the six selected transform coefficients, F(0,0), F(0,1), F(1,0), F(1,1), F(2,1) and F(1,2), are sufficient to give a subjectively acceptable image. Section 9.1 has looked into the classification of blocks into simple shapes. The amount of AC

energy in specific transform coefficients is used to classify the blocks into flat, horizontal, vertical or diagonal blocks.

The results could be used in an adaptive error concealment algorithm. This algorithm will classify the neighbouring blocks into simple shapes and using this information, a selected number of neighbouring blocks and transform coefficients would be used to generate the linear interpolated replacement block. Some future developments are suggested in the next chapter but in many aspects, initial foundation work has been completed. Further improvements to the algorithms are possible but beyond the scope of this work.

# CHAPTER 10 ▷ ▷ ▷



# CHAPTER 10 : FURTHER DEVELOPMENTS AND CONCLUSIONS.

The research was conducted along the lines set out in chapter 1. The work and results were reported in detail in chapter 2 to 9. This last chapter provides suggestions for possible extension of the present work and finally a general discussion and conclusion of each of the main parts in this research is given in the last section.

## **10.1 FURTHER DEVELOPMENTS.**

Possible further developments in fractal-based image coding and error concealment algorithms are given in this section.

# 10.1.1 Fractal-based Image Coding.

A number of further developments are suggested in the last section of chapter 4. The main suggestions include possible shift of sub-blocks' geometry from square to other shapes which should reduce blockiness. Another suggestion is to move from block-based fractal coding to feature-based fractal coding.

## 10.1.2 Additional Side Information.

At the transmitter, DCT blocks can be classifed into a limited set of shapes and texture. Additional side information derived from this classification could be transmitted in separate cells and this information would be useful at the receiver in assisting the error concealment algorithms. The disadvantage in transmitting additional information is to increase the bandwidth requirements, thereby increasing the possibility of cell loss, and would normally not be recommended.



# 10.1.3 Adaptive Error Concealment Algorithms.

At the receiver, additional information can be derived from the neighbouring blocks surrounding the lost block. Each neighbouring block can be classified into a fixed number of simple shapes. If a horizontal structure is identified on both the left and right neighbouring blocks, it would be sensible to assume that the lost block is more likely to have a horizontal structure and only the left and right blocks would be necessary to generate the replacement block using linear interpolation. A vertical structure along the top and bottom neighbouring blocks would be a strong indication that the lost block has a vertical structure. Similarly for the diagonal structure. If structures are not prominent in the neighbouring blocks, replacement would be estimated from the linear interpolation of the four neighbouring blocks.

The error concealment algorithms would thus be adaptive, and only the necessary blocks would be used to generate the replacement for the lost block.

# 10.1.4 Other Applications of Error Concealment Algorithms.

The error concealment algorithms investigated in detail during this research could be implemented on other DCT-based compression algorithms. It would be interesting to implement some sort of error concealment algorithm on MPEG or H.261 video stream. Error concealment algorithms studied could be used for the correction of errors in the DCT-based intra-frames within the video stream.

# 10.2 SUMMARY AND CONCLUSIONS.

There are two aspects which are of special interest in this work. The initial work concentrates on an alternative image compression algorithm but the main bulk of this work is the development of error concealment algorithms for the treatment of video cell lost.

A number of image compression techniques assume a wide sense stationary autoregressive image model and redundancy in such images is exploited to achieve compression. An imaginative compression technique generating a lot of interest is the use of self-similar fractals, in which a search is made for self-similar structures within an image and describing these structures by a set of equations. For images with recognisable objects that exhibit a large degree of self similarity, very high compression ratios are achievable. Images generated from this method look similar to a graphics constructions. Fractal based compression methods for natural images have been reported in the technical press and a particular compression algorithm centred on block-based fractal coding was investigated in chapter 3 and 4.

At the encoder, each image was divided into range blocks and larger domain blocks. For every range block in the image, a search was conducted within the image for a matching domain block. When a match was found, the required contractive affine transformation for describing the transformation from the matching domain block to the range block was recorded. The set of contractive affine transformations found for the whole image was used to uniquely described this image at the decoder. At the decoder, an initial image was generated from the mean value of each range block and the set of contractive affine transformations was applied to the initial image. After one or two iterations, a good enough replica of the original image was available.

Occam programs for the encoding and decoding of images were implemented on a transputer-based display system. A set of  $256 \times 256$  monochrome images were used to test the algorithm and the results achieved a compression ratio of 0.8-1.3 bit/pixel for this set of images. Images after compression and decompression were subjectively acceptable, texture and edges were well preserved but some blockiness was visible. This compression ratio is comparable to standard compression algorithms, like those based on the DCT, but is much more computionally intensive particularly for coding. An alternative compression algorithm with much higher compression ratio was not identified.

An improvement to the resilience of image transmission over a packet network exhibiting cell loss was achieved using error concealment algorithms. Error concealment methods for DCT-based images were investigated in chapter 6 and these included :

- Replacing the area where the error occurs with a grey block, or a block of a suitably chosen colour.
- Replacement by the previous frame for correlated image sequences.
- Replacement by the previous block within the same frame.
- Replacement by linear interpolation of transform coefficients in the neighbouring blocks.
- Replacement by quadratic interpolation of transform coefficients in the neighbouring blocks.

Images after error concealment were significantly better and the most promising method was by linear interpolation of transform coefficients in the neighbouring blocks. Each error was concealed with a replacement block estimated from the neighbouring blocks and the replacements were usually good estimates of the missing block.

A statistical study of the transform coefficients was conducted in Chapter 7 and it provided a better understanding of the relationship and properties of transform coefficients. The study used a set of four images and concentrated on the pixel domain, DC coefficient F(0,0) and first three AC transform coefficients F(0,1), F(1,0) and F(1,1).

The plots of the pixel distributions and DC coefficients' distributions were quite similar suggesting that there was a strong correlation between adjacent pixels. The Central Limit Theorem states that the distribution of a large group of independent random variables will be Gaussian as the distributions of AC coefficients tend to be Gaussian. This suggested that the AC coefficients were uncorrelated but a 2dimensional histogram of a 3-dimensional image array (x and y direction with pixel intensity) would not detect correlation within specific directions. Therefore, another section studied the relationship between transform coefficients in either the horizontal, vertical or diagonal direction.

DC coefficients were, as expected, found to be highly correlated in the horizontal, vertical and diagonal directions. Correlation coefficients for successive AC coefficients F(0,1) in the vertical direction were significant but only in the vertical direction whereas AC coefficients F(1,0) were significant only in the horizontal direction. No significant correlation in any direction was shown for AC coefficients F(1,1) in any of the test images. These results were all justified by reference to the basis images for the particular coefficients.

These findings were used in a modified version of the error concealment algorithm described in Chapter 8, in which the linear interpolation method of neighbouring blocks used only a subset of the transform coefficients. There were no changes to the method of obtaining the DC and F(1,1) transform coefficients but to generate transform coefficient F(0,1), only the corresponding coefficients in the top and bottom blocks were used and for coefficient F(1,0) only the same coefficients in the left and right blocks were used in the interpolation equation.

A reduction in the computational complexity of the algorithm was achieved and a version of the modified error concealment algorithm was tested on the MASTER (Appendix C) platform. An Occam program was written to run as a single process



Chapter 9 described some subjective and objective measurements on image impairment due to the omission of selected transform coefficients. If only a limited number of transform coefficients were used to represent the image, the six selected transform coefficients, F(0,0), F(0,1), F(1,0), F(1,1), F(2,1) and F(1,2), were shown to be sufficient to give a subjectively acceptable image. Another section looked into classification of blocks into simple shapes. The amount of AC energy in specific transform coefficients was used to classify blocks into flat, horizontal, vertical or diagonal blocks. Results from this chapter could be used in a possible adaptive error concealment algorithm. This algorithm would classify the neighbouring blocks into simple shapes and based on this information, a selected number of neighbouring blocks could be used to generate the linear interpolated replacement block.

As shown in this thesis, the investigation was conducted along the research objectives set up in chapter 1. The alternative image compression algorithm using block-based fractal coding achieved an acceptable but not extremely high compression ratio. The main area of research was to improve the resilence of an image transmission system over an ATM network subjected to error and a number of error concealment algorithms were suggested and investigated in this research. The method of error concealment in the transform domain by linear interpolation of the neighbouring blocks showed promising results and the error concealment algorithms investigated in this research could be considered for implementation in DCT-based image transmission systems.

# APPENDICES ▷ ▷ ▷



# **APPENDIX A : DISCRETE COSINE TRANSFORM.**

The primary purpose of any sub-optimum transformation is to convert statistically dependent picture elements into "independent" coefficients. Most of the transformation used are linear and unitary and some considerations for the choice of transformation are as follows:

- Transformation on blocks neglects the redundancies that exist between blocks, therefore on a statistical basis, it is advantageous to have a large subpicture. However for implementation simplicity as well as to exploit local changes in picture statistics and visual fidelity, a smaller subpicture is desirable.
- ii) Compression results from dropping the transformation coefficients with small energy, it is therefore desirable to have a transform which compacts most of the image energy in as few coefficients as possible.
- iii) Another consideration is the ease of performing the transformation itself which will have a direct effect on the computational time required.

The Discrete Cosine Transform (DCT) is a very widely used tool in image compression and this section will give a brief description of how it works, which will in turn explain the functionality of other sub-optimum unitary transform as the basic principle is similar. The  $N \times N$  cosine transform has excellent energy compaction for highly correlated data and it is very close to the K-L optimum transform for a first order stationary Markov sequence. [10]

The image for compression is subdivided into a number of smaller blocks, generally eight by eight pixels, and the two-dimensional DCT is applied. It is a computationally intensive process but is well suited to digital signal processing techniques, it consists of a series of multiplication and accumulation process. The

equations for a two-dimensional DCT and IDCT are shown below and a diagram of the matrices is shown in figure A.1.

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

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



DCT basically converts the spatial information into its equivalent frequency components. The data input in this block (pixel values) are converted into an output block with the same dimension of coefficients, representing the frequency components.

The set of coefficients obtained in this way all have a strong similarity in that there are only a few coefficients with a significant value and that these are concentrated predominantly in the top left-hand corner of the block. The value in this corner (top left) is known as the DC coefficient and is a measure of the average value of the pixels in the block; the rest are known as the AC coefficients, and their frequency increases the further they are from the DC coefficient. If the inverse transform, known as the inverse DCT or IDCT, is now applied to this set of coefficients, the original block of pixel values is recovered.



# APPENDIX B : STATISTICAL PROPERTIES.

### **B.1 STATISTICAL IMAGE MODELS.**

Statistical Image Models are often used in Digital Image Processing. The image is usually treated as a first order autoregressive process (sometimes called the Markov process) with wide sense stationary properties. Parameters of a wide sense stationary processes are readily specified, being the mean and autocorrelation function.

Such images work well for transform coding where because of their statistical properties, the significance of images with properties of an autoregressive process may be carried over in a rather straightforward way from the data (spatial) domain to the transform domain. As has been stated data compression is then attain by removing the non essential components in the transform domain.

#### **B.1.1 First-Order Autoregressive Process.**

Different types of stochastic processes are useful as models for time series. An important class of stochastic processes is those which are stationary. An autoregressive process is just one of the many types of stochastic processes.

The value of a first order autoregressive process [AR(1)] at a particular time, or point in space, depends on the previous value plus an independent random variable.

 $X(t) = \rho X(t-1) + Z(t)$ 

where 
$$Z(t)$$
 is an independent Gaussian sample.

A formal definition of the process is given in Papoulis. [51].



The mean of a 2-dimensional image field of width N and height M is given by the spatial average of the picture element values.

$$\overline{x} = \frac{1}{N \times M} \left[ \sum_{n=1}^{N} \sum_{m=1}^{M} x_{nm} \right]$$

The variance of the image field is the average value of the square of the difference between the value of an arbitrary image element and the mean.

$$\sigma^{2} = \frac{1}{N \times M} \left[ \sum_{n=1}^{N} \sum_{m=1}^{M} (x_{m} - \overline{x})^{2} \right]$$

#### **B.1.3 1-Dimensional** *k*-step Covariance & Correlation Coefficient.

Restricting the treatment to 1-dimensional, the inter-element relationships are investigated. Consider first a single line of the scanned image containing elements  $x_1, x_2, ..., x_M$ . If;

$$a = x_1, x_2, x_3, \dots, x_{M-1}$$
  
$$b = x_2, x_3, x_4, \dots, x_M.$$

calculating the pointwise products and average, we get;

$$\sigma_{1,2}^2 = \sigma_{2,1}^2 = \frac{1}{M-1} E(a-\overline{a})(b-\overline{b})$$

Assume that  $\overline{a} = \overline{b} = \overline{x}$ , the overall mean,

$$\sigma_{1,2}^2 = \sigma_{2,1}^2 = \frac{1}{M-1} \sum_{m=1}^{M-1} (x_m - \overline{x})(x_{m+1} - \overline{x})$$

Therefore, in general, the 'kth step' covariance of the sequence is defined as;

$$\sigma_{1,k+1}^2 = \sigma_{k+1,1}^2 = \frac{1}{M-k} \sum_{m=1}^{M-k} (x_m - \overline{x})(x_{m+k} - \overline{x})$$

by normalising with the overall variance, it gives the k-step correlation coefficient.

$$C(k) = \rho_k = \frac{\sigma_{kl}^2}{\sigma^2}$$

#### **B.1.4 Model of Correlation Coefficient in a Image Field.**

For Image Models (with first order autoregressive process), [10] it is quite frequently considered a suitable model for an image field to relate the correlation coefficient at step k to the coefficient at step 1 by

$$C(k) = \rho_k = \rho^{|k|}$$

This method is not too inaccurate for the correlation displacement relationship for images which are 'smooth', contains relatively little image details.

#### **B.2 INTERPRETING THE CORRELOGRAM.**

This section is extracted from the book "The Analysis of Time Series: An Introduction (Second Edition)" by C. Chatfield. [52]

A useful aid in interpreting a set of correlation coefficients is a graph called a correlogram in which the correlation coefficients are plotted against the lag k. There is usually little point in calculating correlation coefficients for values of lag k greater than about N/4, where N is the number of observations and some general advice on interpreting the Correlogram is given in the next page:

#### **B.2.1 A Random Series.**

If a time series is completely random, then for large N, the correlation coefficient  $\rho \equiv 0$  for all non-zero values of k. If a time series is random, the  $\rho_k$  is approximately normally distributed with  $N(0, \frac{1}{N})$ . For a random time series, 19 out of 20 of the values of  $\rho_k$  can be expected to lie between  $\pm 2/\sqrt{N}$ . Therefore if one plots the first 20 values of  $\rho_k$ , then it can expect to find one 'significant' value on average even when the time series really is random.

#### **B.2.2** Short-term Correlation.

Stationary series often exhibit short-term correlation characterised by a fairly large value of  $\rho_1$  followed by 2 or 3 more coefficients which, while significantly greater than zero, tend to get successively smaller. Values of  $\rho_k$  for longer lags tend to be approximately zero. An example of such a correlogram is shown in figure B.1. A time series which gives rise to such a correlogram, is one for which an observation above the mean tends to be followed by one or more further observations above the mean, and similarly for observations below the mean. A model called an autoregressive model, may be appropriate for series of this type.



172



If a time series has a tendency to alternate, with successive observations on different sides of the overall mean, then the correlogram also tends to alternate. The value of  $\rho_1$  will be negative. However the value of  $\rho_2$  will be positive as observations at lag 2 will tend to be on the same side of the mean. A typical alternating time series together with its correlogram is shown in fig B.2.



## **B.2.4** Non Stationary Series.

If a time series contains a trend, then the values of  $\rho_k$  will not come down to zero except for very large values of the lag. This is because an observation on one side of the overall mean tends to be followed by a large number of further observations on the same side of the mean because of the trend. A typical nonstationary time series together with its correlogram is shown in figure B.3. Little can be inferred from a correlogram of this type as the trend dominates all other features. In fact the sample autocorrelation function should only be calculated for stationary time series and so any trend should be removed before calculating the correlation coefficients.





# **APPENDIX C : THE MASTER PROJECT.**

# C.1 THE MASTER PROJECT.

This section describes the implementation of the final MASTER workstation.

#### C.1.1 Equipments.

#### Basis Workstation.

The IBM PC compatible computer forms the basis of the MASTER workstation, additional adapter cards provide multimedia and networking facilities. The PC runs MS-DOS and Microsoft Windows v3.1 in 386 enhanced mode. The adapter cards used to provide multimedia capture and play back facilities are made by VideoLogic. See figure 6.1. These cards were chosen by the project because of their relative low-cost at the time, the symmetric form of the video compression algorithm used, and the supportive nature of the suppliers.

#### Network Devices.

As stated earlier the project used MATMX to provide wide area connection over ISDN. The local interface to MATMX is the transputer link. Networking facilities were provided by a set of transputers mounted in TRAMS on a Transtech TMB16 TRAM motherboard.

#### Multimedia Capture and Playback Devices.

The VideoLogic DVA-4000 is a frame store device that accepts several video sources as input. The DVA-4000 combines the video generated images with the normal graphical video output from the PC. It can capture frames individually and save them to disk. Alternatively, the DVA-4000 can pass sequences of



Figure C.1: The MASTER Project

frames to an accompanying adapter card, the MediaSpace for compression and storage.

The VideoLogic MediaSpace card incorporates a C-Cube CL550 processor that compresses video frames using Motion-JPEG, and is powerful enough to encode 25 frames per second of video. The MediaSpace card also contains several DSP chips to perform audio record and play back. The process that runs on the T400 transputer on the MediaSpace, known as the IOP, handles the transfer of data between the PC and the encoding hardware. The IOP can be instructed to send and receive multimedia data through an external transputer link rather than the PC bus. Using this facility continuous transfer of data from the camera to the network, and from the network to the display, is independent of the main PC memory, bus and processor. This allows users to continue using the workstation for other applications whilst a videoconference is taking place.

Videoconferencing requires the presence of two video and audio data streams: A locally generated stream for transmission to remote sites; and a remotely generated stream for local display. It requires two complete sets of VideoLogic card to support these two data streams and thus a workstation requires two PCs. Within a single workstation these PCs are called the transmit (or Tx) PC and the receive (or Rx) PC. In order to keep within the strict philosophy of a single PC

or workstation the transmit PC is used only to generate a local video and audio data stream with all the workstation functionality running on the receive PC. The project could dispense with the transmit PC should full duplex multimedia communication hardware become available in the future.

On the Rx PC, as far as the MediaSpace and DVA-4000 cards are concerned only one image is ever displayed. The two data streams, i.e., local image and the remote image(s) are joined together by the processes on the network card. These cards allowed the project to attain an acceptable quality audio and video. Audio was sampled at 11.025 KHz and encoded using 8 bits PCM. The frame rate was 25 frames per second, with a window size of around 256 by 144 pixels on a VGA (640x480) screen. This size provides a picture aspect ratio of 16 by 9 which produces a more suitable picture for conferencing, allowing two people to sit side-by-side comfortably and still fill the frame.

# C.1.2 PC Process Architecture.

The processes that run on the PC are control related and can be logically divided into, those that control the setup of the multimedia hardware and those that handle network communication. Controlling the whole process is the MASTER user interface program (MASTWCUI).

# PC Multimedia Control.

The VideoLogic MIC II system provides simple control over multimedia functionality, allowing a Windows application to send command scripts to the hardware in a straight forward manner. These are interpreted by the MIC II software, which in turn sends commands either directly to the DVA 4000, or, via the ATP to the MediaSpace. The ATP (Asynchronous Transport Process) controls the MediaSpace card and transfer of multimedia data between the disk and the card.

To implement MASTER it was necessary to write a windows application to control the presentation and capture of the multimedia data stream (MASTWCUI), and also make modifications to the ATP.

MASTWCUI first initialises the hardware and displays a window on the screen. The parameters for hardware initialisation are controlled from a pre-recorded VideoLogic Movie file. A local video image is shown in a quadrant on the screen. When the user becomes a member of a conference this picture is joined by a copy of the remote video stream.

# PC Network Control



Fig C.2: The Process Architecture

In order to pass messages from MASTWCUI to the network it was necessary to write the following code:

- 1 MASTDOS, a Windows VDM (Virtual Dos Machine) to communicate with the transputers.
- 2 MASTVXD, a Windows VxD (Virtual Device Driver) to pass messages between MASTDOS and MASTWCUI.
- 3 MASTAPI, a Windows DLL (Dynamic Link Library) to provide an API (Application Programmer Interface) to the VxD.

The MASTDOS code that polls the TMB16 on a regular basis to see if a message is available. It was necessary to develop this as a VDM, because Windows can only pre-emptively multi-task VDMs and not Windows applications. This VDM runs at a low priority, so most of the system resources are still given to windows applications. The integration of the PC processes and the hardware is shown in figure 6.2.

This mechanism allows messages to be passed from MASTWCUI to processes running on transputers.

# C.1.3 Transputer Processes Architecture.

The transputer processes are arranged in the following configuration over four TRAMS.



Fig C.3: The Transputer Process Architecture

## Transmit MediaSpace Control.

When the transmit PC is ready, its MediaSpace sends a data packet to the Tx.MediaSpace control process. This process receives this packet and then continually asks for video and audio packets. A video packet is asked for every 40 milliseconds and audio every 46 milliseconds. These delays were chosen to give 25 frames per second of video and a roughly constant audio packet size. Packets are then sent both to the bandwidth management process and to the joining process.

## Bandwidth Management.

A simple scheme is implemented, where audio and video packets are stored separately in two small buffers, each buffer holding two packets. When the

# Segmentation and Reasssembly (SAR).

The SAR scheme used is based on AAL5 (ATM Adaptation Layer). The data part of each cell carries a portion of the packet. After transmission of full cells, a final data cell carries the remaining data. A special "end-of-packet" cell follows, identifiable by a special header bit being set, which contains the length of the packet in the data field. The AAL5 checksum is not implemented.

After each transmission of a packet the segmentation process waits for a length of time, dependant on both the size of the packet and the bandwidth of the link, before receiving a new packet from the bandwidth management process. This exerts back-pressure forcing discard of packets.

The reassembly process reconstructs packets from the incoming cell stream, maintaining a separate buffer for each virtual channel. On receipt of an "end-of-packet" cell the packet is forwarded to the concealment process, together with a count of the number of cells used to reconstruct the packet, and the length in the final cell.

# Error Concealment.

The error concealment process receives a reassembled frame and checks to see if its length matches the number of cells used to reconstruct it. If a small number of cells are missing then the frame is examined and missing MCU blocks replaced. Of the schemes described in the error concealment chapter, interpolation of above and below coefficients and a fixed colour replacement block schemes have been implemented. If a larger number of cells are missing then the entire frame is dropped.

# Multipoint Signalling.

The multipoint signalling process controls the workstation. When the user elects to join a conference at a remote site a virtual channel is established for MASTER signalling. A join message is then sent to the remote site. If the joining is accepted, then a data channel is set up to transmit both continuous and conventional data.

For a third or fourth party to join the conference they must make a join request to the conference owner. If the join is accepted then the conference owner informs the new participant which other participants are in the conference. The new party now issues link messages to the other participants. These are similar to join messages except that connection is automatically established.

When a multipart conference is taking place a link consisting of 2 virtual channels (data and signalling) exists between each two parties.

## Video & Audio Joining.

Joining of both video and audio streams is performed at the packet level. The transputer maintains four buffers, one for each of the remote streams and one for the local stream.

# APPENDIX D : LIST OF PUBLICATIONS.

#### **PUBLICATIONS:**

- L.T.Chia, D.J.Parish, J.W.R.Griffiths, I.W.Phillips, G.T.Jones, "On the use of Transform Domain Information for Concealment of Errors in JPEG Images", EUSIPCO-94, VII European Signal Processing Conference in Edinburgh, Scotland, Sept. 1994.
- (2) L.T.Chia, J.W.R.Griffiths, D.J.Parish, I.W.Phillips, "Error Concealment Algorithms for an ATM Videoconferencing System.", TENCON '94, IEEE International Conference on Frontiers of Computer Technology, Singapore, Aug. 1994.
- (3) L.T.Chia, D.J.Parish, J.W.R.Griffiths, "On the Treatment of Video Cell Loss in the Transmission of Motion-JPEG and JPEG Images", *Computer & Graphics, Vol. 18, No. 1, pp 11-20*, Pergamon Press, Jan./Feb. 1994.
- (4) L.T.Chia, D.J.Parish, P.J.Coventry, I.W.Phillips, J.W.R.Griffiths, "Motion-JPEG on a Network and the Treatment of Video Cell Loss" VISICOM'93, 5th International Workshop on Packet Video in Berlin, Germany, Apr. 1993.
- (5) D.J.Parish, P.J.Coventry, I.W.Phillips, L.T.Chia, J.W.R.Griffiths (LUT) & A.G.Tagg, X.Cao (OBU), "Towards a Multimedia, Multi Site, Interworking Environment - The Master Project", *Published as a contribution to a book*.

- (6) D.J.Parish, P.J.Coventry, L.T.Chia, J.W.R.Griffiths, I.W.Phillips (LUT) & A.G.Tagg, X.Cao (OBU), "The MASTER Project: A Multimedia, Multi Site, Interworking Environment", *International Symposium on Multimedia Technologies and Future Applications* in Southampton, U.K, Apr. 1993.
- (7) L.T.Chia, D.J.Parish, J.W.R.Griffiths, P.J.Coventry, I.W.Phillips, "On the Treatment of Video Cell Loss in the Transmission of JPEG Images", *Multi-*Service Networks '93 at Cosener, U.K, Jul. 1993.
- (8) J.W.R.Griffiths, G.J.Lu, S.Jayasinghe, L.T.Chia., "Multimedia Communications in a Medical Environment", SICON'91 in Singapore, Sept. 1991.
- (9) J.W.R.Griffiths, L.T.Chia, G.J.Lu, "Image Compression using Iterated Function Systems" *Transputer Applications 91* in Glasgow, U.K, Aug. 1991.

#### **PRESENTATIONS**:

- "The MASTER Project A Multimedia Workstation for Videoconferencing" - I.W.Phillips, D.J.Parish, J.W.R.Griffiths, L.T.Chia (LUT) & A.G.Tagg, D.H.Shrimpton, X.G.Cao (OBU), 1994 UK IT Forum (DTI - JFIT) in Heriot Watt University, Edinburgh, U.K, Mar. 1994.
- (2) "The MASTER Project" I.W.Phillips, D.J.Parish, J.W.R.Griffiths, L.T.Chia (LUT) & A.G.Tagg, D.H.Shrimpton, X.G.Cao (OBU), S.E.R.C. Multiservice Networks Seminar in Lancaster, U.K, Sept. 1993.

# **REFERENCES.**

- [1] CCITT Recommendation H.261, "Video Codec for Audio-Visual Services at p\*64 kbit/s", Dec. 1990.
- [2] (JPEG1) ISO DIS 10918-1, "Digital Compression and Coding of Continuous-tone Still Images." ISO/IEC JTC1/SC2/WG10, Jan 1992.
- [3] (MPEG) ISO DIS 11172-2 Part 2: Video, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s", ISO-IEC/JTC1 SC29/WG11, Mar. 1992.
- [4] W. Verbiest, L. Pinnoo, B. Voeten, "The impact of the ATM Concept on Video Coding", *IEEE Journal Select. Areas Commun.*, Vol.6, No.9, pp 1623-1632, Dec. 1988.
- [5] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM Networks", *IEEE Journal Select. Areas Commun.*, Vol.7, No.5, pp 801-806, Jun. 1989.
- [6] A.N. Netravali, J.O. Limb, "Picture Coding: A Review", Proc. of the IEEE, Vol.68, No.3, pp 366-406, Mar. 1980.
- [7] A.K. Jain, "Image Data Compression: A Review", Proc of the IEEE, Vol.69, No.3, pp 349-389, Mar. 1981.
- [8] M. Kunt, M. Kocher, A. Ikonomopoulos, "Second-Generation Image-Coding Techniques", Proc. of the IEEE, Vol.73, No.4, pp 549-575, Apr. 1985.
- [9] R. Forchheimer, T. Kronander, "Image Coding From Waveforms to Animation", IEEE Trans. Acoust. Speech Signal Processing, Vol. ASSP-37, No.12, pp 2008-2023, Dec. 1989.
- [10] R.J. Clarke, "Transform Coding of Images", Academic Press, London, 1985.
- [11] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantiser Design", IEEE Trans. Commun, COM-28, pp 84-95, Jan. 1980.

- [12] N.M. Nasrabadi, R.A. King, "Image Coding using Vector Quantisation: A Review", IEEE Trans. Commun, COM-36, pp 957-971, Aug. 1988.
- [13] M. Vetterli, "Multi-Dimensional Sub-Band Coding : Some Theory and Algorithms", Signal Processing, Vol.6, No. 2, pp 97-112, Apr. 1984.
- [14] J.W. Woods, S.D.O'Neil, "Subband Coding of Images", IEEE Trans. Acoust., Speech, Signal Processing, Vol.ASSP-34, pp 1278-1288, Oct. 1986.
- [15] H. Gharavi, A. Tabatabai, "Sub-Band Coding of Monochrome and Color Images", IEEE Trans. Circuits Syst., Vol.35, No.2, pp 207-214, Feb. 1988.
- [16] S.G. Mallat, "Multifrequency Channel Decompositions of Images and Wavelet Models", *IEEE Trans. Acoust., Speech, Signal Processing, Vol.ASSP-37, pp 2091-2110,* 1989.
- [17] J. Froment, S. Mallat, "Second Generation Compact Image Coding with Wavelets", Wavelets- A Tutorial in Theory and Applications (Ed. C.K.Chui), pp 655-678, 1992.
- [18] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, "Image Compression Using Wavelet Transform", *IEEE Trans. Image Processing, Vol. 1, No. 2,* pp 205-220, Apr. 1992.
- [19] C.S. Choi, H. Harashima, T. Takebe, "Analysis and Synthesis of Facial Expressions in Knowledge-Based Coding of Facial Image Sequences", *Proc. ICASSP, M9.7, pp 2737-2740*, May 1991.
- [20] G.K. Wallace, "Overview of the JPEG (ISO/CCITT) Still Image Compression Standard", SPIE, Vol. 1244, Image Processing Algorithms and Techniques, pp 220-233, Aug.1990.
- [21] A. Leger, T. Omachi, G.K. Wallace, "JPEG Still Picture Compression Algorithm", Optical Engineering, Vol.30, No.7, pp 947-954, July 1991.
- [22] M.F. Barnsley, "Fractal Everywhere", Academic Press, 1988.
- [23] B.B. Mandelbrot, "The Fractal Geometry of Nature", W.H.Freeman and Co., San Francisco, 1977.
- [24] R.F. Voss, "1/f noise and fractals in DNA base sequences", Applications of Fractals and Chaos, British Computer Society, Feb. 1992.
- [25] L.F. Jardine, I.R. Whitworth, "Fractal Based Synthesis of Visual Texture", *IEE Colloquium on the Application of Fractal techniques in image* processing, Dec. 1990.
- [26] J.C. Vassilicos, F. Tata, "Is there Chaos in Economic Time Series? A study of the Stock and the Foreign Exchange Markets", *Applications of Fractals* and Chaos, British Computer Society, Feb. 1992.

- [27] R.J. Clarke, L.M. Linnett, "Fractal and image representation", IEE Electronics and Communication Engineering Journal, Vol.5, No.4, Aug. 1993.
- [28] H. Peitgen, D. Saupe, "The Science of Fractal Images", Springer-Verlag, 1988.
- [29] M.F. Barnsley, "A Better Way to Compress Images", BYTE magazine, Jan. 1988.
- [30] A.E. Jacquin, "A Novel Fractal Block coding Technique for Digital Images", *IEEE International Conference on Acoustic Speech and Signal Processing, Proc. ICASSP, M8.2, pp 2225-2228, 1990.*
- [31] A.E. Jacquin, "A Fractal Theory of Iterated Markov operators with applications to Digital Image Coding", *Ph.D Thesis, Georgia Institute of Technology*, 1989.
- [32] J.M. Beaumont, "Advances in Block Based Fractal coding of still images", *IEE colloquium on the application of Fractal Techniques in Image Processing*, Dec. 1990.
- [33] J. Waite, "A review of Iterated Function Systems Theory for Image Compression", *IEE colloquium on The Application of Fractal Techniques in Image Processing*, Dec. 1990.
- [34] B. Ramamurthi, A. Gersho, "Classified Vector Quantization of Images", IEEE Trans. Commun, COM-34, No. 11, pp 1105-1115, Nov 1986.
- [35] M. Jeffrey, "A brief Introduction to ATM External Version", Multi-Service Networks'92, Cosener's House, Abingdon, 1992.
- [36] M de Prycker, "Asychronous Transfer Mode Solution for BroadBand ISDN", Ellis Horwood Limited, Chichester, UK, 1991.
- [37] Y. Wang, Q. F. Zhu, "Signal Loss Recovery in DCT-based Image and Video Codecs", Visual Communications and Image Processing' 91, SPIE Vol. 1605, pp 667-678, 1991.
- [38] R. Veldhuis, "Restoration of Lost Samples in Digital Signals", Prentice Hall International, Exeter, UK, 1990.
- [39] "MIC System II." Publication by VideoLogic Limited., VideoLogic Limited, Home Park Estate, Kings Langley, Hertfortshire, WD4 8LZ, United Kingdom.
- [40] D.L. MacAdam, "Visual sensitivities to color differences in daylight", Journal of the Optical Society of America, 32, No.5, pp 247-274, May 1942.

- [41] C.F. Gerald, R.O. Wheatley, "Applied Numerical Analysis (Fourth Edition)", Prentice Hall International, 1990.
- [42] M. Ghanbari, D. Pearson, "Probability Density Functions for Hadamard Coefficients in Transformed Television Pictures", *Electronic Letters*, *Vol.14*, pp 252-254, 1978.
- [43] D.J. Parish, P.J. Coventry, L.T. Chia, J.W.R. Griffiths, I.W. Phillips (LUT) & A.G. Tagg, X. Cao (OBU), "The MASTER Project: A Multimedia, Multi Site, Interworking Environment", *International Symposium on Multimedia Technologies and Future Applications*, Southampton, U.K, Apr. 1993.
- [44] I.W. Phillips, D.J. Parish, J.W.R. Griffiths, L.T. Chia (LUT) & A.G. Tagg, D.H. Shrimpton, X.G. Cao (OBU), "The MASTER Project - A Multimedia Workstation for Videoconferencing" - 1994 UK IT Forum (DTI - JFIT), Heriot Watt University, Edinburgh, U.K, Mar. 1994.
- [45] J.W.R. Griffiths, G.J. Lu, L.T. Chia, S. Jayasinghe, "MultiMedia Communications in a Medical Environment", SICON'91, Singapore, Sept 1991.
- [46] J.J. DePalma and E.M. Lowry, "Sine-Wave Response of the Visual System. II. Sine-Wave and Square-Wave Contrast Sensitivity", J. Opt. Soc. Amer., Vol. 52, pp 328-335, Mar. 1962.
- [47] D.H. Kelly, "Visual Contrast Sensitivity", Opt. Acta, Vol. 24, pp 107-130, Feb. 1977.
- [48] J.L. Mannos and D.J. Sakrison, "The Effect of a Visual Fidelity Criterion on Encoding of Images", *IEEE Trans. Inform. Theory, Vol. IT-20, pp 525-536*, Jul. 1974.
- [49] N.B. Nill, "A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment", *IEEE Trans. Commun., Vol. COM-*33, pp 551-557, Jun. 1985.
- [50] J. Allnatt, "Transmitted Picture Assessment", John Wiley & Sons, 1983.
- [51] A. Papoulis, "Probability, Random Variables and Stochastic Processes (3rd Edition)", McGraw Hill, London, 1991.
- [52] C.Chatfield, "The Analysis of Time Series: An Introduction", Chapman and Hall, London, 1980.

. . .

.