



Pilkington Library

Author/Filing Title JONES

Vol. No. Class Mark T

**Please note that fines are charged on ALL
overdue items.**

LOAN COPY

0402292014



**CONCEALMENT ALGORITHMS FOR
NETWORKED VIDEO TRANSMISSION
SYSTEMS**

by

**Gareth Tudor Jones
B. Sc. M. Sc.**

Supervisors

Dr. D. J. Parish and the late Prof. J. W. R. Griffiths


A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

of the Loughborough University

© by Gareth Tudor Jones, September 1999

 Loughborough University Public Library
Date <i>Out 10</i>
Class
Acc No. <i>40 229 201</i>

MOVO 2333 LB

ABSTRACT.

This thesis addresses the problem of cell loss when transmitting video data over an ATM network. Cell loss causes sections of an image to be lost or discarded in the interconnecting nodes between the transmitting and receiving locations.

The method used to combat this problem is to use a technique called Error Concealment, where the lost sections of an image are replaced with approximations derived from the information in the surrounding areas to the error. This technique does not require any additional encoding, as used by Error Correction. Conventional techniques conceal from within the pixel domain, but require a large amount of processing ($2N^2$ up to $20N^2$) where N is the dimension of an $N \times N$ square block. Also, previous work at Loughborough used Linear Interpolation in the transform domain, which required much less processing, to conceal the error.

The work in this thesis has concentrated on concealing in the transform domain, but with the inclusion of Feature Detection, Classification and Concealment. After initial work on the frequency/transform domain content of various edges, a simple Feature Detector was found, that used the magnitude of certain coefficients, to detect simple edges. However, this technique only detected the edges with a large edge contrast,

also miss-classification occurred. Therefore, a new method to detect features was developed that looked for a features distinctive fingerprint from the ratio of certain low frequency coefficients. Specifically, only three ratios were calculated and from these, edges with varying angles and position could be detected and classified. A number of classifiers were then produced that had an increasing number of diagonal classes from 2 to 4. The edges were then classified using these classifiers, using the ratio information alone. Once this information was found a concealment stage analysed it and calculated if an edge in a surrounding block would enter the error block. If the edge entered the error block, the algorithm calculated the new transform domain coefficients required, from knowledge of how the coefficients vary as an edge is displaced across a block. Again, the ratio information was used to help calculate the new coefficients along with a new fourth ratio. Only 4 coefficients were calculated to keep the algorithm computationally simple, so that concealment could be performed in real time.

The resulting algorithm uses at most 1 addition, 4 subtractions and 3 multiplications in the concealment stage and 3 divisions and a number of comparisons in the detection and classification stages. In addition, this amount of processing load is not proportional to the size of image block used, as is the case with pixel domain concealment methods. An improvement in performance over the Linear Interpolation and Initial Feature detection and Concealment was observed and the algorithm compares favourably to simple pixel domain techniques.

Keywords: Error Concealment Algorithms, ATM Networks, Cell Loss, MPEG, JPEG, MJPEG, Image Transmission, Video Transmission.

ACKNOWLEDGEMENTS.

I would like to thank my supervisor, Dr. D. J. Parish for his continual guidance, support and especially, understanding of my specific situation and limitations, due to my medical history. In addition, I would like to express my gratitude to the late Professor J. W. R. Griffiths for his invaluable academic insight that helped develop the initial stages of this research.

Thanks must also go to Dr. I. W. Phillips and Dr. L. T. Chia for their previous work on the subject, at Loughborough Universities High Speed Networks Laboratory. I would also like to thank the Electronic & Electrical Engineering Department at Loughborough for the financial assistance received, in the form of a Departmental Grant.

Special thanks must go to my Mother and Father for their unconditional financial and moral support, encouragement and belief in my abilities throughout the duration of my research. Thanks also to my sister Beth, for her encouragement and for allowing me to stay at her house during my final weeks at Loughborough, and to Sylvia for putting up with my absence from home for all these years.

GLOSSARY OF ABBREVIATIONS AND SYMBOLS:

- AAL : Asynchronous Transfer Mode Adaptation Layer.
- ATM : Asynchronous Transfer Mode.
- ARQ : Automatic Repeat Request.
- B-ISDN : Broadband Integrated Services Digital Network.
- BCH Codes : Error Detection/Correction code, named after Bose, Chaudhuri and Hocquenqhem.
- CBP : Coded Block Pattern, defined in the MPEG and H.261 standards.
- CCITT : International Telegraph and Telephone Consultative Committee.
- CD-ROM : Compact Disc Read Only Memory.
- CLP : Cell Loss Priority. Single bit in the ATM Header.
- CRC : Cyclic Redundancy Check Code, type of error detection/correction.
- CSPDU : Convergence Sublayer Protocol Data Unit, used within AAL protocols.
- DCT : Discrete Cosine Transform.
- DC, AC : Transform Domain Coefficients, DC is the first coefficient that gives an indication of the average value of a pixel block. The AC

coefficients give an indication of the edge content of a pixel block. $F(0,0)$ is the DC coefficient, all other $F(N,N)$ are the AC coefficients, where $N \times N$ is the dimension of the block.

DEC	: DCT coefficient based spatial error concealment.
DM	: Delta Modulation, a form of waveform coding.
DPCM	: Differential Pulse Code Modulation.
DFT	: Discrete Fourier Transform.
DWHT	: Discrete Walsh Hadamard Transform.
$F(u,v)$: 2-dimensional representation of transform coefficients.
$f(m,n)$: 2-dimensional representation of a pixel block.
FEC	: Forward Error Correction.
FFT	: Fast Fourier Transform, a fast method of calculating the DFT.
GFC	: Generic Flow Control.
GOB	: Group of Blocks, defined in the H.261 standard.
GOP	: Group of Pictures, defined in the MPEG standard.
H.261	: CCITT recommendation for videoconferencing.
HEC	: Header Error Correction.
HDTV	: High Definition Television.
IDCT	: Inverse DCT.
IFFT	: Inverse FFT.
ISDN	: Integrated Services Digital Network.
ISO	: International Standards Organisation.
JPEG	: Joint Photographic Expert Group, image compression technique.
KLT	: Karhunen-Loeve Transform.
LEC	: Linear Interpolation error concealment.
MB	: Macro Block, defined in the MPEG standard.
MCU	: Minimum Coded Unit, defined in the JPEG standard.
MJPEG	: Motion JPEG, used to encode video using the JPEG standard.
MPEG	: Motion Picture Expert Group.
NMSE	: Normalised Mean Squared Error, an objective measurement often used to assess image quality.
PCM	: Pulse Code Modulation.
PCI	: Protocol Connection Identifier.

PSNR	: Peak Signal to Noise Ratio, an objective measurement often used to assess image quality.
PSC	: Picture Start Code.
PDU	: Protocol Data Unit.
PTI	: Payload Type Identifier, in ATM header.
QT's	: Quantisation Tables.
RGB	: Red, Green and Blue colour space.
RLE	: Run Length Encoding.
RTSP	: Real Time Streaming Protocol.
RS Code	: Reed-Solomon error correction code.
SAR	: Segmentation and Reassembly Layer.
SSC	: Slice Start Code.
STM	: Synchronous Transfer Mode.
TCP	: Transfer Control Protocol as used with Internetworking.
TFDCEC	: Transform Domain Detection Classification and Error Concealment.
UNI	: User Network Interface.
VBR	: Variable Bit Rate.
VCI	: Virtual Channel Identifier.
VC	: Virtual Channel.
VP	: Virtual Path.
VPI	: Virtual Path Identifier.
VLC	: Variable Length Encoding.
VQ	: Vector Quantisation.
YUV, YCrCb	: Luminance and Chrominance colour space.

TABLE OF CONTENTS.

	Page
Abstract	i
Certificate of Originality	iii
Acknowledgements	iv
Glossary of Abbreviations and Symbols	v
Table of Contents	vii
1. Introduction	1
1.1 Research Objectives	3
1.2 Thesis Organisation	3
2. Image Processing	5
2.1 Image Compression Techniques	6
2.2 Waveform Coding	7
2.2.1 Pulse Code Modulation	7
2.2.2 Delta Modulation	8
2.2.3 Differential Pulse Code Modulation	9
2.3 Transform Coding	9
2.3.1 Karhunen-Loeve Transform	9
2.3.2 Wavelet Transform	10
2.4 Image Model Coding	11

2.4.1 Fractal Image Compression	12
2.5 Quantisation	12
2.5.1 Scalar Quantisation	12
2.5.2 Vector Quantisation	12
2.6 Codeword Assignment	13
2.7 Standardised Image Compression Schemes	14
2.7.1 The JPEG Compression Algorithm	14
2.7.1.1 JPEG File Structure	17
2.7.2 The H.261 Compression Algorithm	19
2.7.2.1 Motion Estimation	23
2.7.2.2 H.261 File Structure	24
2.7.3 The MPEG Compression Algorithm	25
2.7.3.1 MPEG File Structure	28
2.8 Summary and Conclusions	29
3. Transmission of Digital Video across a Computer Network	30
3.1 ATM Cell Structure	32
3.2 ATM Adaptation Layer	34
3.3 ATM Cell Loss	35
3.3.1 Error Correction	36
3.3.1.1 Data Retransmission	36
3.3.1.2 Forward Error Correction	36
3.3.2 Error Concealment	37
3.3.2.1 Pixel Domain Error Concealment	37
3.3.2.2 Transform Domain Error Concealment	38
3.3.2.3 Interleaving and Cell Packing Strategies	39
3.4 Summary and Conclusions	40
4. Transform Domain Feature Detection and Block Classification	41
4.1 Edge Classification	42
4.2 Initial Classification	42
4.3 Advanced Classification	48
4.4 Classification Results	70
4.5 Summary and Conclusions	71
5. Transform Domain Error Concealment	72
5.1 Transform Domain Coefficient Calculation	94
5.1.1 DC00(n)	94
5.1.2 AC01(n) and AC10(n)	95
5.1.3 AC11(n)	100
5.1.4 All Remaining Coefficients	102
5.2 Results and Performance Evaluation	108
5.3 Summary and Conclusions	120
6. Further Developments and Conclusions	123
6.1 Further Developments	123
6.2 Summary and Conclusions	126
Appendix A: The Discrete Cosine Transform	129
References	132

CHAPTER 1

CHAPTER 1:

INTRODUCTION

The general thrust of development in modern communications is towards increasing digitisation. For example, the widespread use of computers has led to an increased need for man-computer and computer-computer communications and thus an increasing proportion of the information we wish to transmit is inherently digital. There are many reasons for this, for example, digital signals provide very low error rates in transmission, and such errors as do occur may be detected and/or corrected or concealed by using suitable data encoding. Also, digital signals are compatible with digital logic circuits and computer systems and can thus be economically processed and re-routed as required using such circuits and systems.

The economic benefits of digital switching and signal processing provide a major justification for the use of digital transmission for multimedia data in an Integrated Services Digital Network (ISDN/B-ISDN). The switching protocol standard chosen for implementing the Broadband Integrated Services Digital Network is ATM

(Asynchronous Transfer Mode) [1,2], which is a highly flexible multiplexing and network access protocol supporting a wide range of data, voice and video services. ATM allows the transmission of video and image data at variable bit rates giving substantial bandwidth saving, constant picture quality and the ability to allocate variable bandwidth to different services. Cell delay jitter and cell losses are the major disadvantages to transporting images over an ATM network. Cell delay jitter occurs due to the non-uniform arrival of the cells at the receiver, as a result of random queuing delays in the network. Delay jitter in packet video is not considered to be as serious a problem as in packet voice transmission, however cell loss which can cause impairments in images and video and the possible loss of synchronisation, is a serious problem. The two main mechanisms of cell loss are bit errors in the header information causing miss-delivery of cells and bursts of losses caused by queue overflow in the event of traffic congestion. The probability of occurrence increases with the bit rate and thus more probable with video services than with audio. Cell loss requires the application of error recovery techniques, which are broadly divided into two main areas i.e. Error Correction and Error Concealment. Error Concealment uses an approximation derived from existing information in the data stream, as opposed to Forward Error Correction [3, 4] which can offer perfect recovery but requires the addition of redundant information into the data stream with a corresponding increase in the transmission bandwidth, complexity and hence cost. In addition, Forward Error Correction and Automatic Retransmission Request schemes (ARQ) have been found to have some limitations, especially in the wireless context [5]. Hierarchical Transmission of coefficients [6, 7] is another unattractive alternative to cell loss recovery. Replacement of the lost information by the same information in the previous frame is not always feasible especially if there has been motion or a scene change.

In order to detect the cell losses both Error Correction and Concealment use a technique that labels the cells at the transmitter and the receiver checks the validity of the received sequence of labels. This is the only redundant information introduced when using Error Concealment whereas Error Correction introduces additional redundant information to perform detection and correction. With a 1.5 Mb/s (average rate) video, the overhead for an MPEG file containing a slice per MCU in I-frames and one slice per P and B-frame would equate to 3 to 4% of extra bandwidth.

However a FEC scheme using a Reed-Solomon code RS (32, 28) with a symbol size equal to 8 bits, as used with Compact Discs Players, with block interleaving requires about 14% of extra bandwidth.

1.1 Research Objectives

This thesis describes research into addressing the problem of data loss due to channel errors, or otherwise, when transmitting compressed video or image data over a data network. Specifically, the transmission of images or video data over ATM networks is investigated, with particular emphasis on simple/fast concealment algorithms so as to be implementable in real time. The techniques and algorithms could also have application in wireless and other types of data networks. The algorithms to be developed should conceal for losses from within the transform domain, to take advantage of the reduction in data to be processed.

1.2 Thesis Organisation

Chapter 1: Gives a brief introduction and description of the problem to be researched, along with the research objectives, and an outline of the thesis is given.

Chapter 2: Describes the main image processing features/components used within this research along with a general background of the subject area.

Chapter 3: Gives a brief description of the workings and components of the Asynchronous Transfer Mode of transmission. In addition, the problem of cell loss in such networks is addressed. This includes a brief explanation of the existing methods used to combat such losses.

Chapter 4: Looks into the use of simple and efficient transform domain, feature detection and block classification, by investigating the transform domain content of certain edges/features. This leads on to an initial classifier, followed by a more advanced classifiers using three of the low frequency coefficients to form ratios to classify edges/features into a predetermined number of classes. Three different classifiers are developed and demonstrated, that classify edges without the contrast of

the edges affecting the classification results. This results in a block being placed/classified into a certain class within a given classifier, along with the edge's position within the $N \times N$ sized block.

Chapter 5: Looks into the production of simple and efficient transform domain concealment algorithms, by analysing and modelling the way the coefficient within a block change as an edge is displaced across it. From this knowledge, and the use of the ratios developed in Chapter 4, algorithms are developed that require a minimal amount of processing. In addition, a further ratio is developed to assist concealment, which again is unaffected by the edge contrast of the edges in the surrounding blocks.

Chapter 6: Provides suggestions for future research in the area of error concealment and specifically transform domain error concealment. A summary and conclusion of the research work performed is also given.

Appendix A: Gives an overview of the 2 dimensional, Discrete Cosine Transform (DCT).

CHAPTER 2

CHAPTER 2:

IMAGE PROCESSING.

Image processing is a broad and wide ranging subject, and due to the ever increasing developments in Information Technology, as a result of the advancements in digital electronics, the majority of research is directed towards the increasing introduction and application of digitally processed and coded images. Analogue representations are not as amenable to transmission, analysis and storage, compared to digital representations.

Digital Image Processing includes the Enhancement, Restoration and Coding of a mass of information that can be represented optically. Image Enhancement is the processing of images to improve their appearance to human viewers, as in television. Restoration is the reduction or elimination of degradation e.g. noise or image blurring. A major objective of Coding is to represent an image with as few bits as possible, while preserving the level of quality and intelligibility required by the given application. Image coding has two major application areas. One is the reduction of channel bandwidth required for image transmission systems e.g. video conferencing,

digital television. The other application is the reduction of storage requirements, so that more image information can be saved on a given amount of storage media. Digitised images and video require very high bit rates, to transmit the vast amount of information content within such signals across a network. For example an uncompressed High Definition Television (HDTV) signal requires more than 100 Mbps. Therefore, compression algorithms that reduce the bit rates are of great importance, especially as bandwidth is an expensive and usually limited commodity within a network.

2.1 Image Compression Techniques

As described above there is a need for Image Compression Techniques, especially with the ever-increasing use of the Internet by all sections of the population, be they in the academic community or not. The presence of images and video within web pages is becoming ever popular, and communication between users via a videophone, or videoconference type of arrangement is the next step. Now, the transfer rates available within the Internet are limited, especially if a user is connected via a modem link. Therefore, the more compression that can be achieved, the shorter the time required for transmission and real time communication can become an achievable goal.

There are three major components to image compression, as shown in Figure 2.1 following. The first and most important element is the transformation of the image, from the pixel domain, into a domain that is suitable for quantisation and codeword assignment. There are three broad categories: (i) Waveform Coders, (ii) Transform Coders and (iii) Image Model Coders, which code different aspects of an image. The second element is quantisation: To represent an image with a finite number of bits, the image intensities, transform coefficients or model parameters must be quantised.

The third element in image compression is assignment of codewords, the strings of bits that represent the quantisation levels. Each component in Figure 2.1 attempts to exploit the redundancy present in the image source and the limitations of the display device used and the human visual system. Therefore, the elements are closely interrelated.

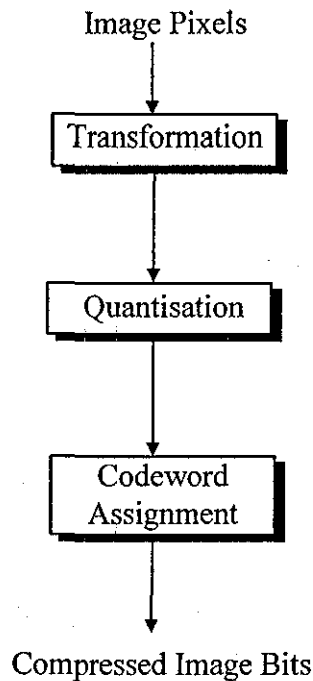


Figure 2.1

2.2 Waveform Coding

In Waveform coding, the image intensity or some simple variation is coded, which allows the coders to be simple both conceptually and computationally. In situations where high quality images are required, the bit rate reduction achievable is comparable to that possible with transform coders. However, when bit rate reduction is important at the expense of a certain amount of quality, then Waveform coders do not perform as well as transform coders.

2.2.1 Pulse Code Modulation

Pulse code modulation (PCM) is the simplest form of waveform coding, where the image intensity $f(n_1, n_2)$ is sampled and then quantised by a uniform quantiser, to produce a digital image. If each sample is quantised by a quantiser having 256 levels, then each pixel is represented by 8 bits and for most applications sufficient quality and intelligibility is preserved. Now, if bit rate reduction is required, the amount of levels

within the quantiser must be reduced, giving a smaller amount of bits/pixel. When the bit rate falls to about 5 to 6 bits/pixel the quantisation noise begins to be visible as random noise and below 3 to 4 bits/pixel, the quantisation noise becomes visible as false contours, due to luminance jumps in the reconstructed image in regions where the original intensity varies slowly. Several simple techniques have been developed to improve the performance of the basic PCM system, some of the main examples are shown in the following section.

2.2.2 Delta Modulation

In the basic PCM system explained above, each pixel is coded in isolation i.e. no reference is made to the previous pixel. That is, the correlation that exists between pixel intensities is not exploited. Delta Modulation (DM) is one method that takes advantage of this correlation. Specifically, the difference between two consecutive pixel intensities is coded by a one-bit quantiser, which produces a signal whose variance is significantly smaller than that of the original pixel intensities. To encode the current sample $f(n)$, the most recently reconstructed sample $f'(n-1)$ is subtracted from $f(n)$. The difference signal $e(n) = f(n) - f'(n-1)$ is quantised to $Y/2$ if $e(n)$ is positive and $-Y/2$ if negative, where Y is the step size. The difference signal $e(n)$ quantised by the one bit quantiser is denoted by $e'(n)$, and $e'(n)$ is sent at the transmitter. At the receiver, $e'(n)$ is added to $f'(n-1)$ to obtain $f'(n)$. The step size Y is an important design parameter. When there are regions in an image where the input samples vary slowly, the reconstructed signal varies rapidly around the original. This is called Granular Noise, and a large Y results in a large amount of granular noise, so a small Y is desired. However, when the samples increase or decrease rapidly it may take many samples before $f'(n)$ can catch up with $f(n)$ using a small Y . This effect is called Slope Overload Distortion, and a large Y is required to reduce this distortion. Therefore, a compromise between the two requirements is required. To obtain a good quality reconstructed image using DM, without significant granular noise or slope overload distortion, a bit rate of 3-4 bits/pixel is required.

2.2.3 Differential Pulse Code Modulation

Differential pulse code modulation (DPCM) can be viewed as a generalisation of DM. The most recently reconstructed sample $f'(n-1)$ can be viewed as a prediction of $f(n)$ and then $e(n)$ can be viewed as the error between $f(n)$ and a prediction of $f(n)$. However, in DPCM a prediction is obtained from more than one previously coded pixel intensity. In addition, in DPCM more than one bit is used within the quantiser to code the error $e(n)$. The same prediction is used at the receiver as that used at the transmitter. Good quality images result when a bit rate of 3 bits/pixel is used.

2.3 Transform Coding

In transform coding, an image is transformed to a domain that is significantly different from the image intensity domain, as used by waveform coders. In applications, where low bit rates below 1 bits/pixel are required e.g. video conferencing, transform coding methods perform significantly better than waveform coding. However, this is at the expense of increased computation and complexity.

There are two main properties observed within transform coding that are utilised to achieve compression. The '*correlation reduction property*', which contributes to the '*energy compaction property*', which results in a large amount of energy being concentrated in a small fraction of the transform coefficients, for typical images [8]. Therefore, if only a fraction of the coefficients are coded i.e. a bit rate of below 1 bit/pixel, there will be a relatively small sacrifice in image quality and intelligibility. Many different transforms have been considered for transform image coding, which differ in energy compaction efficiency and computational requirements.

2.3.1 Karhunen-Loeve Transform

The Karhunen-Loeve transform (KLT) is an optimum transform and is the best of all linear transforms for energy compaction. However, there are serious practical difficulties implementing the KLT, due in part to the fact that there is no computationally efficient algorithm to compute the transform coefficients. This results in a processing load of N^2 by N^2 multiplication's and N^2 by N^2 additions for each N by

N block of pixels. The basis function used in the KLT depends on the covariance function, $K(n_1, n_2; l_1, l_2)$, of the image data $f(n_1, n_2)$. Different images are likely to have different covariance functions, and for each $K(n_1, n_2; l_1, l_2)$, the basis function must be calculated. In addition, the KLT is based on the assumption that $f(n_1, n_2)$ is a random process and its covariance function is known. In practice, the random process assumption for an image is rarely valid, and the covariance function must be calculated/estimated, for each image to be encoded. If the estimate is not accurate, the optimal theoretical properties of the KLT no longer hold.

Therefore, the KLT is rarely used in image processing applications and is replaced by sub-optimum transforms having fixed sets of basis functions. There are a number of sub-optimal transforms including: - Discrete Fourier Transform (DFT) and associated FFT for efficient computation, Discrete Sine Transform, Hadamard Transform, Discrete Walsh Hadamard Transform (DWHT) and the Discrete Cosine Transform whose basis vectors can be shown to be closely related to the basis vectors of the KLT [9], and is addressed further in Appendix A. The DFT, DCT and DWHT are all popular, as fast algorithms exist for their implementation [10].

2.3.2 Wavelet Transform

The FFT, DCT and DWHT highlight all the frequency components at once. No indication is given as to which signal frequencies occur when. To help determine such detail, the signal can be processed using shorter time (or space) segments, limiting the time in which the signal may change. While this improves the time localisation of the spectrum, using shorter time segments degrades the frequency resolution further. Accordingly, if the signal evolves over time, or is simply short lived, the FFT, DCT and DWHT approach has shortcomings.

Wavelet theory offers an alternative way of analysing and manipulating signals. All the benefits of Wavelet theory have yet to be fully gauged, however, those already demonstrated are significant [11]. Wavelets are seen as being particularly suited to data compression, and for the analysis of short duration and wideband signals.

As suggested by the name, a wavelet is a truncated wave, oscillatory in nature and finite in duration. Using the wavelet transform, a signal can be decomposed into a set of related wavelets. Each wavelet element is a scaled version (either compressed or stretched) of a 'mother' wavelet. To determine the contribution or weight of each wavelet, a comparison is performed with the signal as the wavelet is varied continuously in scale (compressed and dilated) and translated in time (or space). The measure of similarity is the weighting, called the wavelet coefficient.

2.4 Image Model Coding

In image model coding, an image is analysed to produce estimated model parameters. These model parameters can then be used to synthesise the original image, and there is potential to synthesise intelligible images at a bit rate substantially lower than that necessary for waveform or transform coders.

Image model coding exploits the fact that synthesising intelligible images does not require accurate reproduction of image intensities. The features of an image essential to image intelligibility are retained and the non-essential features are grossly approximated. For example, many regions within an image are not essential to image intelligibility e.g. background regions having repetitive structure, and such regions can be coded using simple models. Therefore, an image must be segmented into a number of different regions. One method segments an image into a set of contours, and some form of easily modelled texture fills the regions bounded by the contours [12]. The contours can be found by using edge detection techniques on a bi-level representation of the original image.

2.4.1 Fractal Image Compression

Fractal compression can be thought of as a form of Image Model Coding. From an original image a number of mathematical functions that describe a sequence of transformations are repeatedly applied, producing progressively smaller images that eventually converge on a unique fractal [13]. These mathematical functions along with the unique fractal can be thought of as the model parameters.

These methods of image coding are still in the research stage, and require further investigation before their use becomes feasible in very low bit rate applications. However, the cost of very low bit rate coding is increased complexity and processing load, as compared to traditional Waveform and Transform Coding.

2.5 Quantisation

Quantisation is the process of converting a continuous quantity, having an infinite number of possible levels, e.g. pixel intensities, transform coefficients, model parameters e.t.c, into a predetermined, finite number of levels. The fact that any human sense, be it sight or hearing, can detect only finite intensity differences can compensate for the errors introduced by the quantisation process, and in many circumstances be taken advantage of.

2.5.1 Scalar Quantisation

As the name implies, each scalar value is quantised independently with respect to the past or future scalars within a given signal. There are two main types of scalar quantisers i.e. Uniform and Nonuniform quantisers. In Uniform quantisation, the separations between the decision thresholds and the separation between the reconstruction levels of the quantiser have a common value called the *step size*. However, if the signal to be quantised is more likely to be in one particular region than in others, then it is reasonable to assign more reconstruction levels to that region. Quantisation in which reconstruction and decision levels do not have even spacing is called Nonuniform quantisation. In this way, the overall quantisation error between the original scalar values and the quantised values can be reduced.

2.5.2 Vector Quantisation

If two or more scalars are quantised jointly, the procedure is called Vector quantisation or Block quantisation, the major advantage of vector quantisation is its performance improvement over scalar quantisation of a vector source. Now, within vector quantisation an N dimensional vector, consisting of N continuous amplitude

scalars f_i (2.0), are mapped into another N dimensional vector, chosen from L possible reconstruction or quantisation levels (2.1).

$$\mathbf{f} = [f_1, f_2, f_3, \dots, f_N]^T \quad \dots\dots (2.0).$$

$$\mathbf{r} = [r_1, r_2, r_3, \dots, r_N]^T \quad \dots\dots (2.1).$$

$$\mathbf{f}' = \text{VQ}(\mathbf{f}) = \mathbf{r}_i, \quad \mathbf{f} \in C_i \quad \dots\dots (2.2).$$

Where VQ represents the vector quantisation operation, \mathbf{r}_i for $1 \leq i \leq L$ denotes L reconstruction levels, and C_i is called the i th cell. If \mathbf{f} is in cell C_i , \mathbf{f} is mapped to \mathbf{r}_i .

Vector quantisation can lower the average distortion provided by a given number of reconstruction levels as compared to scalar quantisation. Conversely, using vector quantisation can reduce the amount of reconstruction levels required for a given distortion. Vector quantisation achieves this by exploiting the statistical dependence among the scalars within a sequence of a given signal.

2.6 Codeword Assignment

The specific reconstruction level chosen, as a result of quantisation, is required to be transmitted to the receiver. Therefore, a specific codeword is assigned to each reconstruction level, so that the receiver can identify each reconstruction level from its codeword. In addition, since the reconstruction levels are transmitted in sequence, in most cases, the codewords have to be *uniquely decodeable*. There are two methods used for codeword assignment, Uniform Length and Variable Length Codeword Assignment.

Uniform Length codeword assignment is the simplest method of codeword assignment, where each reconstruction level is represented/identified by a fixed length codeword. This method produces an average bit rate that is equal to the bit rate of the individual messages i.e. the reconstruction levels. Therefore, as the number of reconstruction levels used within the quantiser increases, the bit rate of the codewords increases proportionately.

Variable length codeword assignment attempts to take advantage of the statistical occurrence of different message possibilities, to provide an optimum average bit rate code, an example of which is the Huffman Variable Length Code [14]. Those messages that frequently occur are assigned short codewords, and messages that infrequently occur are given long codewords. In this way, although the individual bit rate of the messages can vary widely, the average bit rate for a given sequence of messages can be reduced significantly, compared to the uniform length codeword assignment method.

2.7 Standardised Image Compression Schemes

Because there are a myriad of image compression techniques and a need for images to be encodeable/decodeable on a range, if not all, image compression/decompression hardware and software, International standardisation is required. The three main standards for image compression and decompression are briefly described below.

2.7.1 The JPEG Compression Algorithm

The Joint Photographic Experts Group is an international digital image compression standard for continuous-tone (multilevel) still images, both greyscale and colour [15, 16]. JPEG has three modes, Lossless, Progressive and the Baseline/Sequential mode, which is the mode used within this research, and is briefly described below.

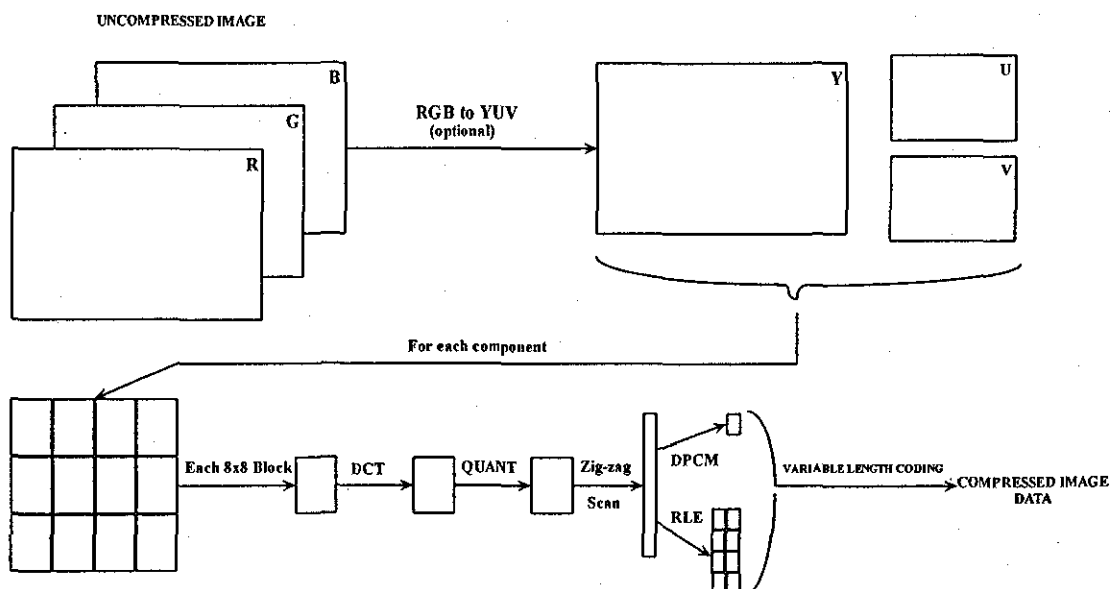


Figure 2.2

As can be seen in Figure 2.2 the first stage is an optional colour space conversion from Red Green Blue (RGB) format to YUV, luminance and chrominance components. This conversion is performed in an attempt to reduce the amount of correlation that exists between the RGB components. The chrominance components are sub-sampled with a corresponding loss in quality. However, the human eye is less sensitive to fluctuations in the colour (UV) than it is to fluctuations in intensity (Y). Therefore, if the sampling ratios are 4:2:2 i.e. for every four blocks of luminance there are 2 blocks each of the chrominance components UV, then the chrominance information is halved and the overall amount of information is reduced by a third.

After this conversion each component is divided into blocks of 8×8 pixels and then the 2 dimensional Discrete Cosine Transform is performed on each block. This produces 64 transform domain coefficients, representing spatial frequency, from 64 pixel values (see Appendix A). The DCT concentrates the energy within the block into a small number of coefficients concentrated in the top left hand corner of the resulting 8×8 coefficient block. This results in a number of coefficients being zero or having very small values.

The following Quantisation stage is designed to discard these coefficients using a Uniform quantiser. This is achieved by dividing each coefficient by a constant, N , and rounding the result. Each transform coefficient is assigned a quantisation constant, which results in a *Quantisation Table*, containing all the constants for all 64 transform coefficients. In addition, the quantisation stage is designed to discard the high frequencies (lower right corner) within the coefficient block, which the eye is less sensitive to than the low frequencies (upper left corner). The JPEG standard defines 2 default quantisation tables, one for luminance and one for chrominance, as shown in Figure 2.3. The next stage in the conversion is to convert the quantised coefficient 8×8 block into a 1×64 vector, where the low frequency coefficients are grouped into the top of the vector. This is achieved by performing a zig-zag scan of the coefficient block, as shown in Figure 2.4. This zig-zag scan is used to increase the run-length of zero coefficients found in the block.

16	11	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	56	24	26	56	99	99	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	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	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	99	99	99	99	99	99	99

Luminance Quantisation Table

Chrominance Quantisation Table

Figure 2.3

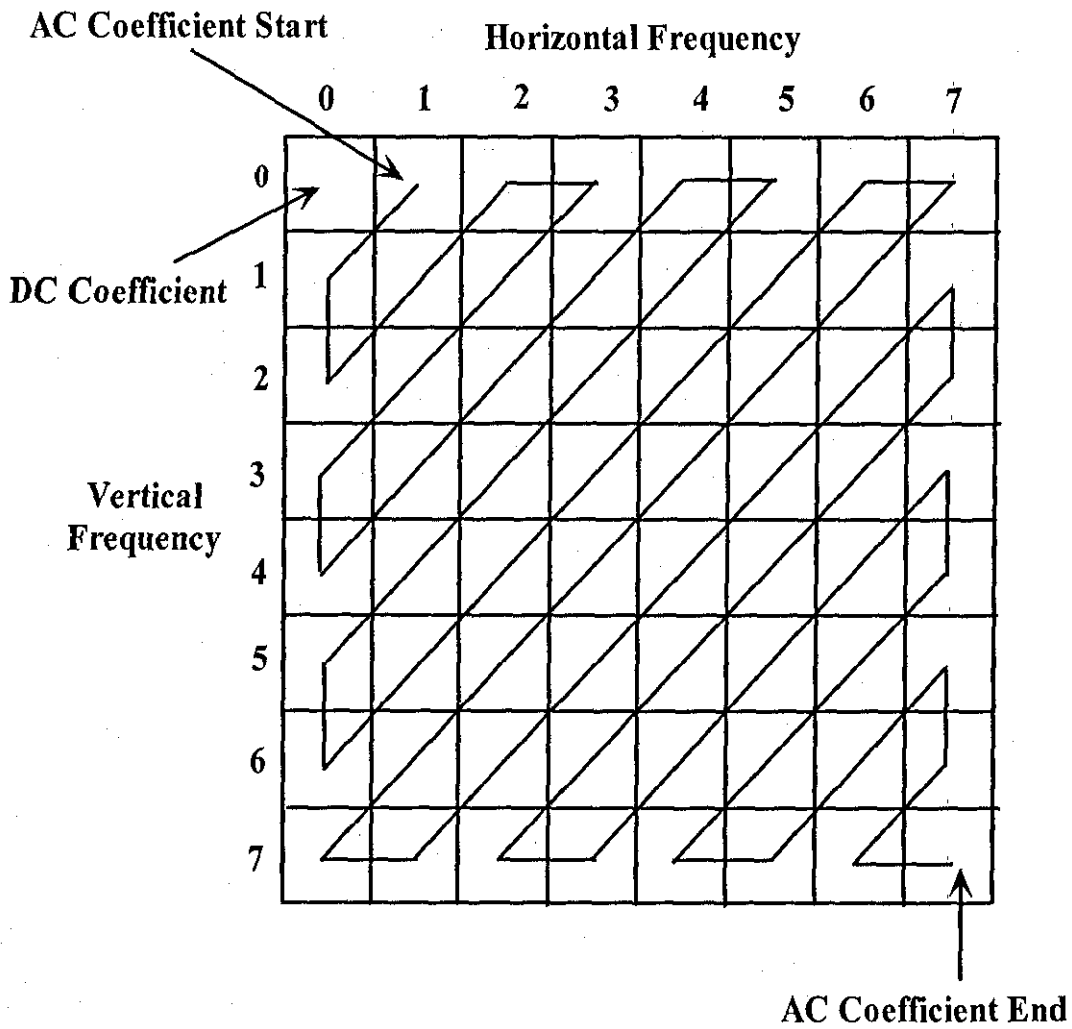


Figure 2.4

The next stage in the process is the DPCM coding of the DC coefficient, where the difference between the DC coefficient of the previous 8×8 block is encoded. This is followed by the Run Length Encoding of the zig-zag scanned 1×64 vector of AC coefficients. The final stage is the Variable Length Encoding, or codeword assignment, where the *Huffman* encoding scheme, developed by David A Huffman in 1952 [14], is used. The Huffman encoding scheme achieves data compression by encoding data based on its frequency of occurrence. The resulting tables can be produced on the fly i.e. custom tables, which are sent in the header information, or the default tables can be used.

The encoding of video information i.e. a number of frames per second, by encoded each frame using the JPEG algorithm is called Motion JPEG. Although video information can be compressed more with the MPEG and the H.261 algorithms (sections 2.7.2 and 2.7.3 following), by taking advantage of the temporal redundancies, Motion JPEG has a number of advantages over these algorithms. The major advantage of Motion JPEG over all varieties of MPEG and H.26x algorithms is the fact that it does not disturb the individual frames in the video sequence. That is, H.261 and MPEG exploit the redundancy in moving pictures by generating P-frames and P-frames and B-frames respectively. Due to the fact that these frames are not real video frames, any attempt to use them as references for editing purposes produces unpredictable results [17], and quality slow motion and rewinds, as used in the editing process, is not available. In addition, a Motion JPEG decoder can also be used as an encoder, this is not the case with an MPEG or H.261 decoder.

2.7.1.1 JPEG File Structure

Figure 2.5 shows the file structure of the JPEG algorithm where the following are: -

- A 'Frame' is a picture, a 'Scan' is a pass through the pixels (e.g., the red component), a 'Segment' is a group of blocks and a 'Block' is an 8×8 group of pixels.

- The Frame Header includes: Sample precision, width, height of image, number of components, unique ID for each component, horizontal/vertical sampling factors for each component and quantisation table to use for each component.
- The Scan Header includes: Number of components in scan, component ID for each component and Huffman table for each component.
- Miscellaneous information that can occur between headers: Quantisation tables, Huffman tables, Arithmetic Coding tables, Comments and Application data.

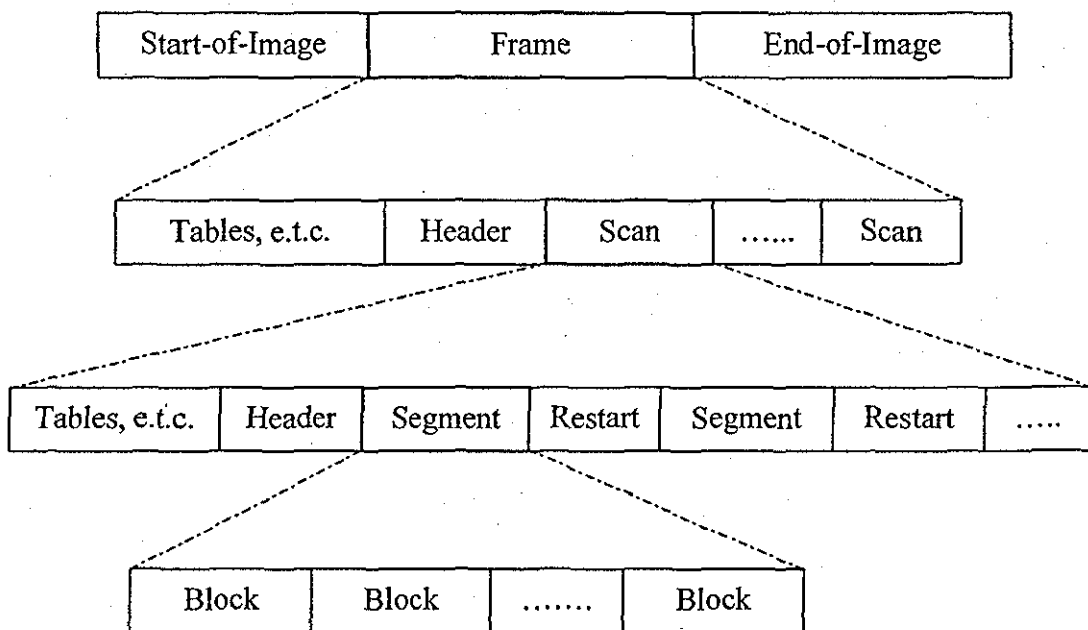


Figure 2.5

2.7.2 The H.261 Compression Algorithm

H.261 was developed by the CCITT for videoconferencing and videotelephone applications over ISDN telephone lines, at $p \times 64$ kbits/sec ($p = 1, 2, 3, \dots, 30$) [18]. H.261 calls for fully encoding some frames and for coding only the difference between a frame and the previous frame in other cases. Motion compensation, which improves image quality, is an option. The two frame types are Intraframes (*I-frames*), and Interframes (*P-frames*). The *I-frames* basically use the JPEG algorithm as seen in section 2.7.1 previously, and *P-frames* use 'pseudo-differences' from a previous frame

("predicted"), so the frames depend on each other. The Macroblocks, where motion compensation is performed, are 16×16 pixels and usually consist of 4 Y luminance blocks, 1 Cr and 1 Cb chrominance blocks. This is a sampling ratio of 4:2:0, as opposed to the 4:2:2 of the JPEG algorithm. Also, the Quantisation process uses a constant value for all DCT coefficients, therefore, no quantisation tables are required. This constant can be changed depending on the 'fullness' of the output buffer i.e if the buffer is becoming full then the quantisation scale factor is increased to reduce the amount of encoded data.

Figure 2.6 shows a block diagram of the encoding process for the H.261 algorithm. The interframe differences are minimised by the use of the motion compensation unit. A macroblock from the current frame is taken, and a best match is searched for from the previous frame, over a region of up to ± 15 pixels in the horizontal and vertical directions. The best matched position is subtracted from the macroblock of the current frame, resulting in a reduced picture difference signal, giving a reduced bit count at the output. The values for the horizontal and vertical translations are included as additional data in the output buffer i.e. the motion vectors.

The motion compensation process is not exact, therefore, the higher frequency components may be badly matched to the current frame data, producing poor predictions. The spatial filter following the motion compensation unit blurs the prediction image data, thereby removing the poor prediction components by low pass filtering.

In situations where a large change in image content has occurred e.g. large amount of motion, or a scene change, the motion compensated prediction is very poor and therefore the macroblock is encoded directly (Intraframe mode).

The variable length codes produced are very sensitive to errors, in an attempt to counteract this a BCH (Bose-Chaudhuri-Hocquenghem) forward error correction encoding scheme is incorporated. Figure 2.7 shows a simple block diagram of the decoding process.

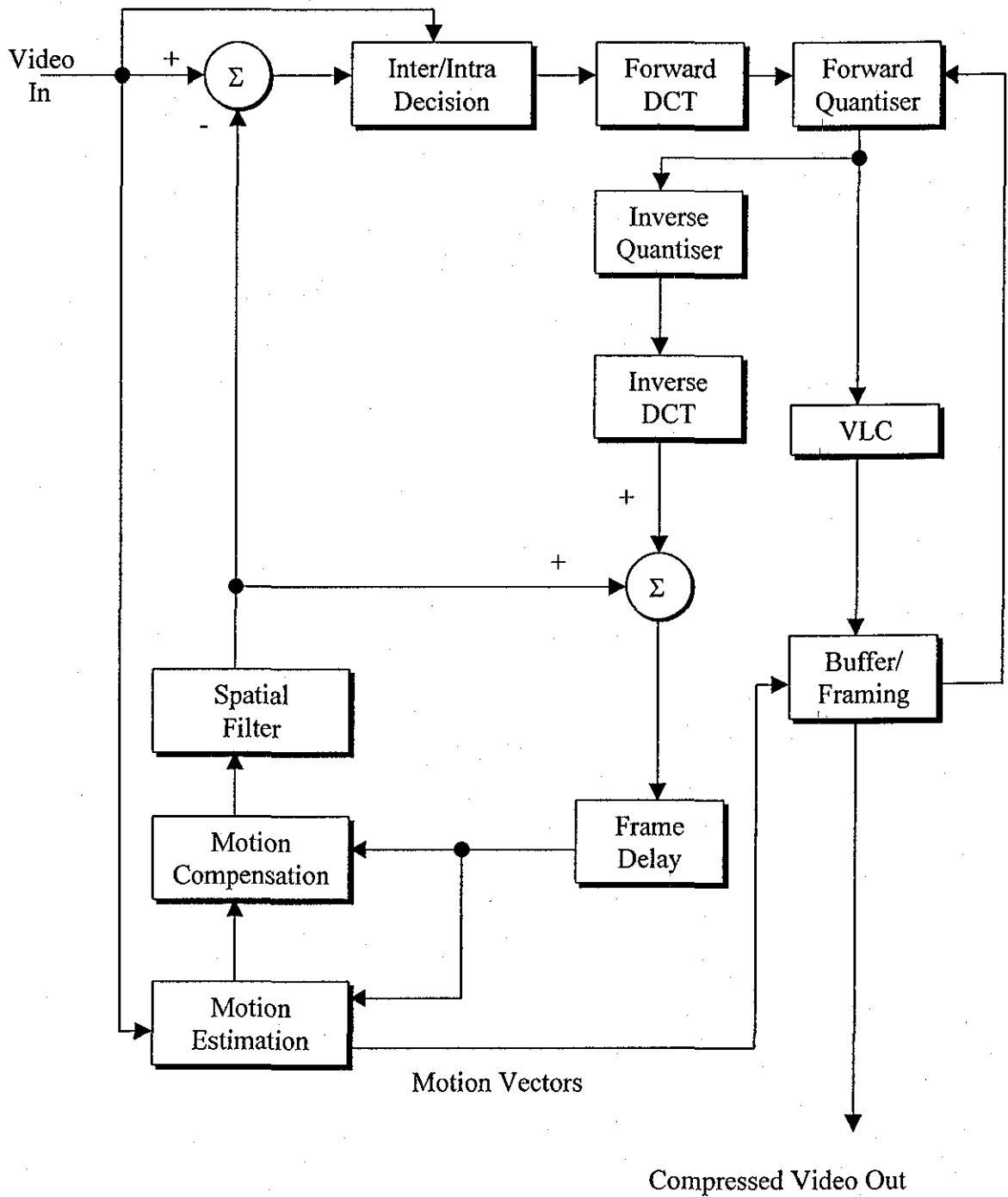


Figure 2.6

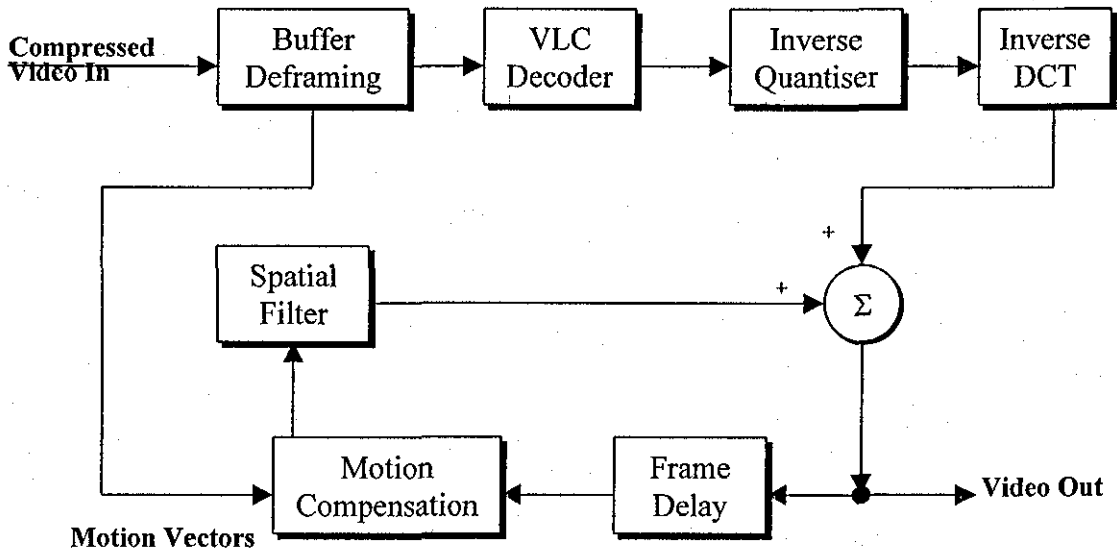


Figure 2.7

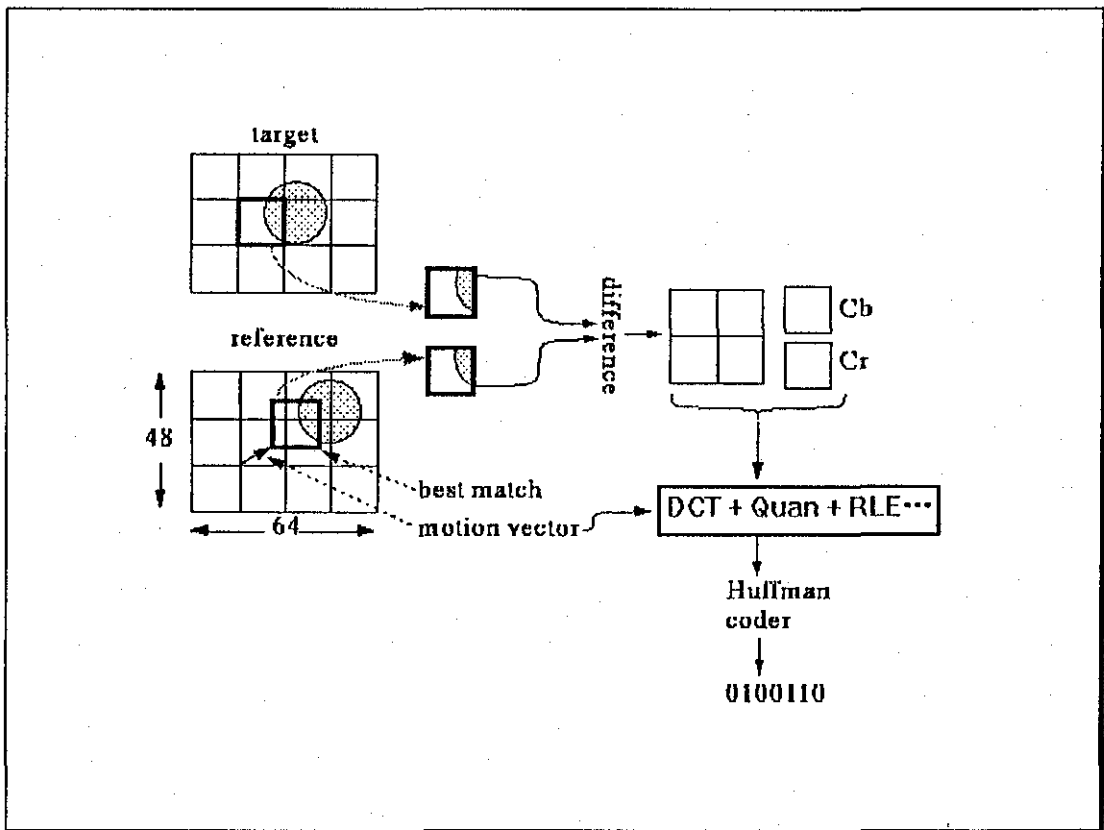


Figure 2.8

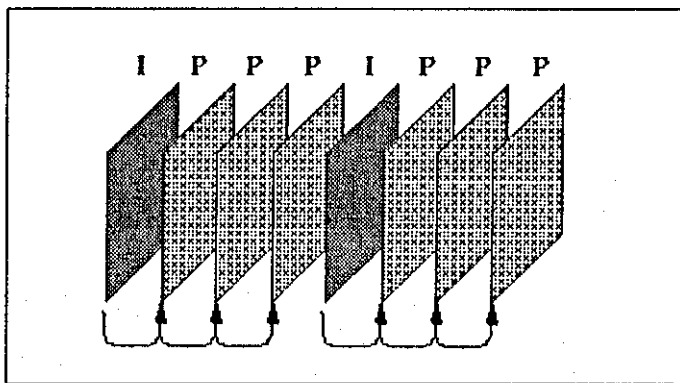


Figure 2.9

Figure 2.8 shows an example of the coding of an Interframe or P frame, where the previous frame is called the *reference* and the current frame the *target*. As can be seen, the differences between the two macroblocks i.e. current macroblock to be encoded and the chosen macroblock from the motion estimation section are DCT encoded, quantised e.t.c. The motion vector for the chosen macroblock from the previous frame is also transmitted, and at the decoder, the group of pixels indicated by the vector are added to the decoded differences to give a good approximation of the current macroblock. Figure 2.9 shows the interdependence of consecutive frames within an H.261 encoded video stream.

2.7.2.1 Motion Estimation

The goal of motion estimation is to find a vector (x, y) such that a given 'Cost function' is minimum. An example of a Cost function is the *Mean Absolute Error* between the pixels in a macroblock in the target frame and pixels in a macroblock in the reference frame. There are a number of methods of searching for the optimum vector within a given region, these include the Full and the Two Dimensional Logarithmic search methods, each having a given processing load and performance. The Full search calculates the Cost function at every possible combination of the motion vector, which is expensive computationally but the performance is optimum. The Two Dimensional Logarithmic method initially calculates the Cost function within a window of $[-p/2, p/2]$ at nine locations, where $[-p, p]$ is the search region. The location that yields the minimum Cost function is used in a new search with half

of the previous size, centred on this location. This process continues until the size of the search region is one pixel wide, when the position with the minimum Cost function is used to calculate the motion vector i.e. the horizontal and vertical translation from the pixel in the target to the pixel chosen within the search. The computational load is greatly reduced, however the performance is suboptimum.

2.7.2.2 H.261 File Structure

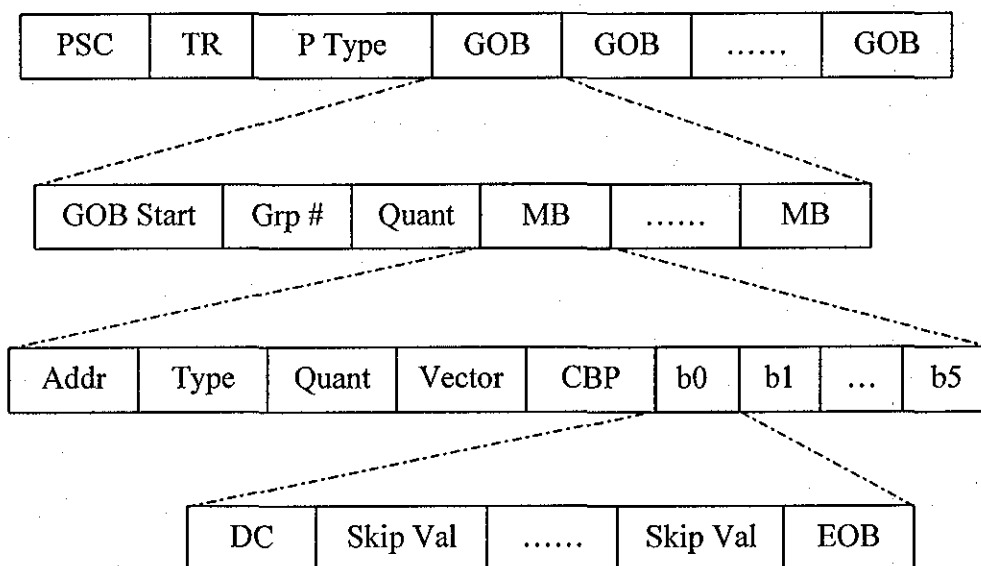


Figure 2.10

Figure 2.10 shows the file structure for the H.261 algorithm where the following are:

- PSC – Picture Start Code, to delineate boundaries between pictures.
- TR – Temporal Reference, timestamp for picture (used for audio synchronisation).
- P Type – Picture Type, to indicate if image is a P-frame or an I-frame.
- GOB – Groups of Blocks, picture divided into regions of 11×3 macroblocks.
- Grp # - Group Number, used if required to skip whole groups.
- MB – Macroblocks.
- Addr – Address of each block in image.

- CBP – Coded Block Pattern, indicates which blocks are present

2.7.3 The MPEG Compression Algorithm

The Moving Pictures Expert Group is a committee that was formed under the auspices of the International Standards Organisation in 1988 to generate standards for digital video and audio compression [19]. Its goal is to achieve a standard that specifies the coded bit stream and decoder requirements for high quality digital and audio. Figures 2.11 and Figure 2.12 following show block diagrams for an MPEG Encoder and Decoder respectively.

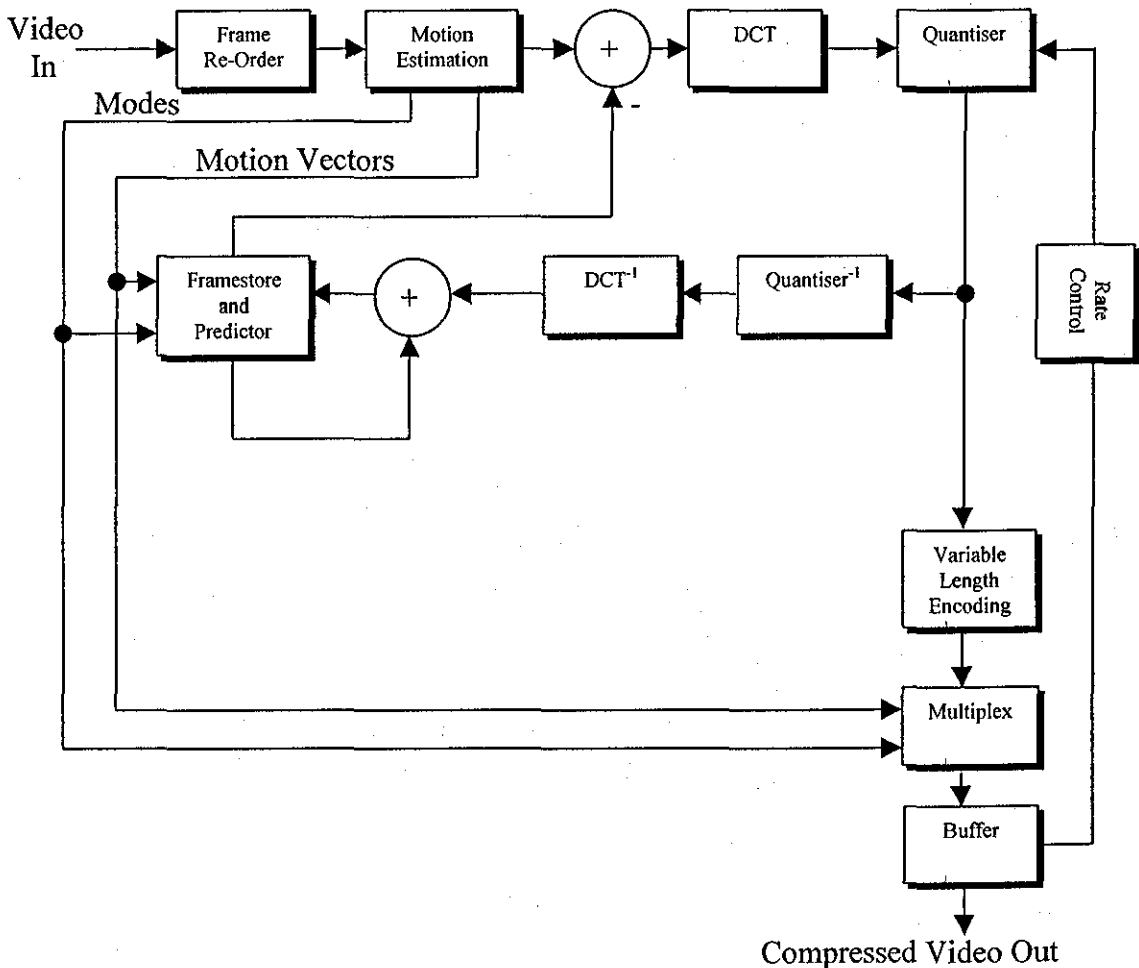


Figure 2.11

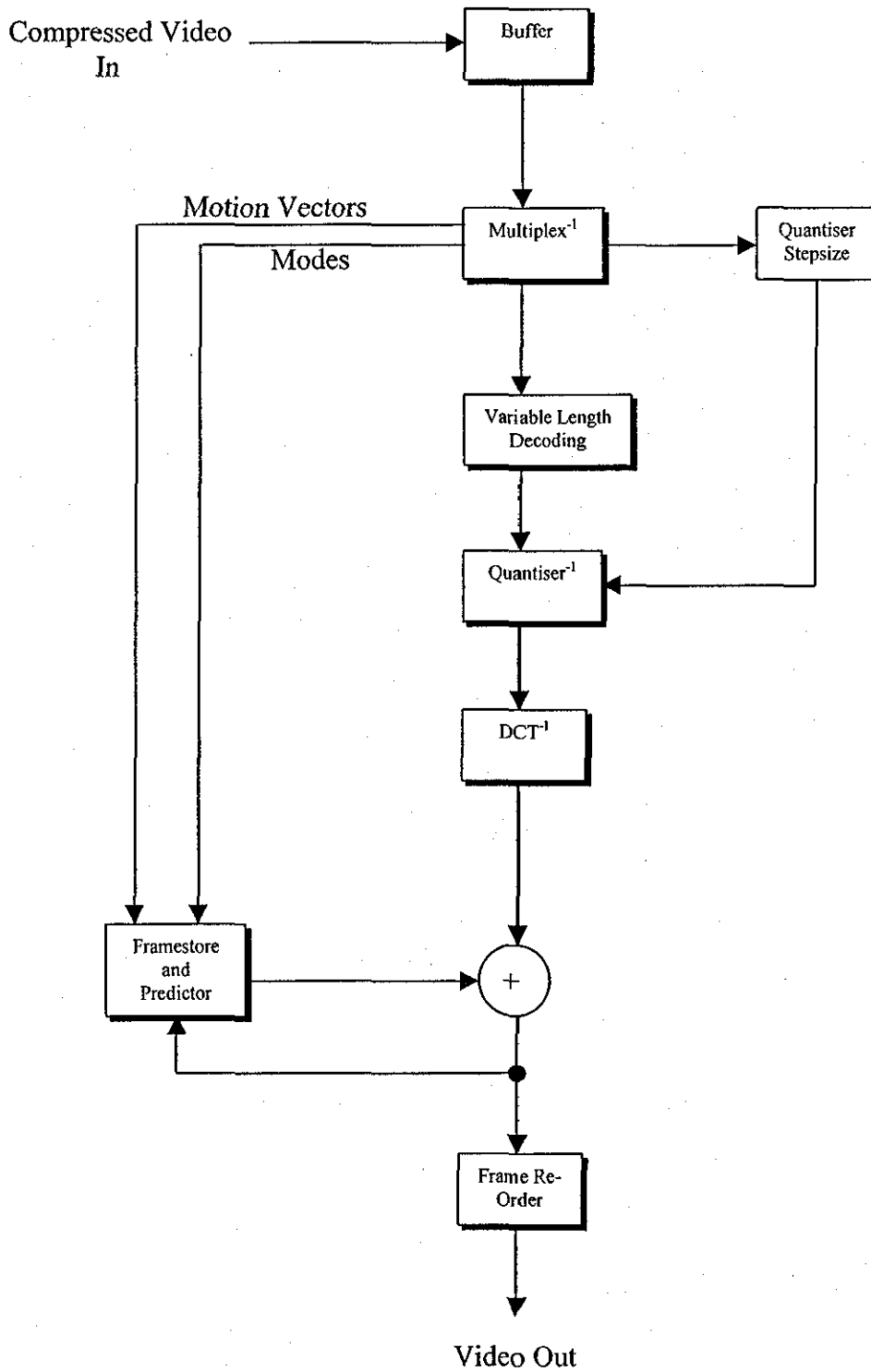


Figure 2.12

MPEG has three frame types, namely Intraframes (*I frames*), Predicted frames (*P frames*) and Bidirectionally predicted frames (*B frames*). I frames are the key reference for the other two frame types and are basically encoded as per the JPEG algorithm, and provide access points for random access. P frames are coded with reference to a past I or P frame and are used as a reference for future predicted frames. B frames, which provide the highest compression ratios, are coded with reference to a past and future reference I or P frame, therefore, B frames are never used as reference frames. Figure 2.13 shows the relationship between the three frames.

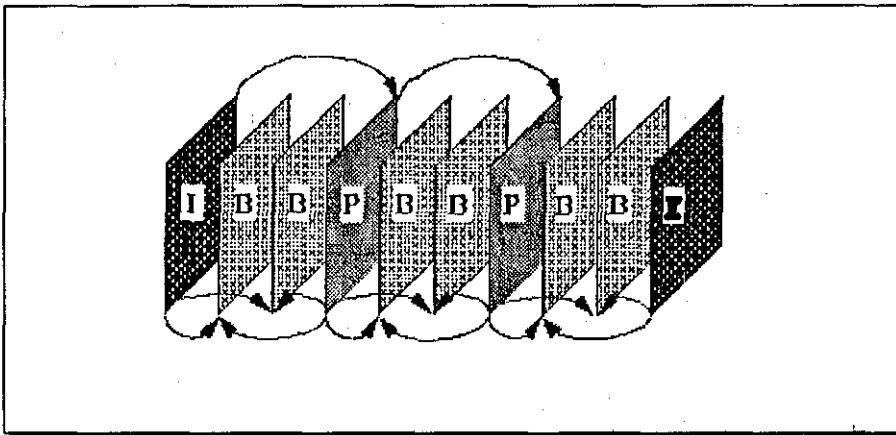


Figure 2.13

The arrows at the top of the picture between I and P frames demonstrate the relationships between frames for forward prediction and the arrows at the bottom of the picture demonstrate the relationships between frames for bidirectional prediction. Motion compensated prediction is used within the P and B frames to improve the compression ratio of the encoded video frames. However, depending on the size of the motion vector a given macroblock can be either Intraframe coded, Interframe coded (forward, backward or average) with motion vectors, or not coded at all if the motion vector is zero. This indicates that the macroblock contains the same information as the corresponding macroblock in a past or future frame. An example of Interframe coding (average) within a B frame is shown in Figure 2.14 following. The past reference and the future reference are interpolated/averaged and the difference between this average and the target or current macroblock is further encoded/compressed by performing the

DCT, Quantisation, Run-length encoding and Variable length encoding. The motion vectors are also differentially encoded before variable length encoding.

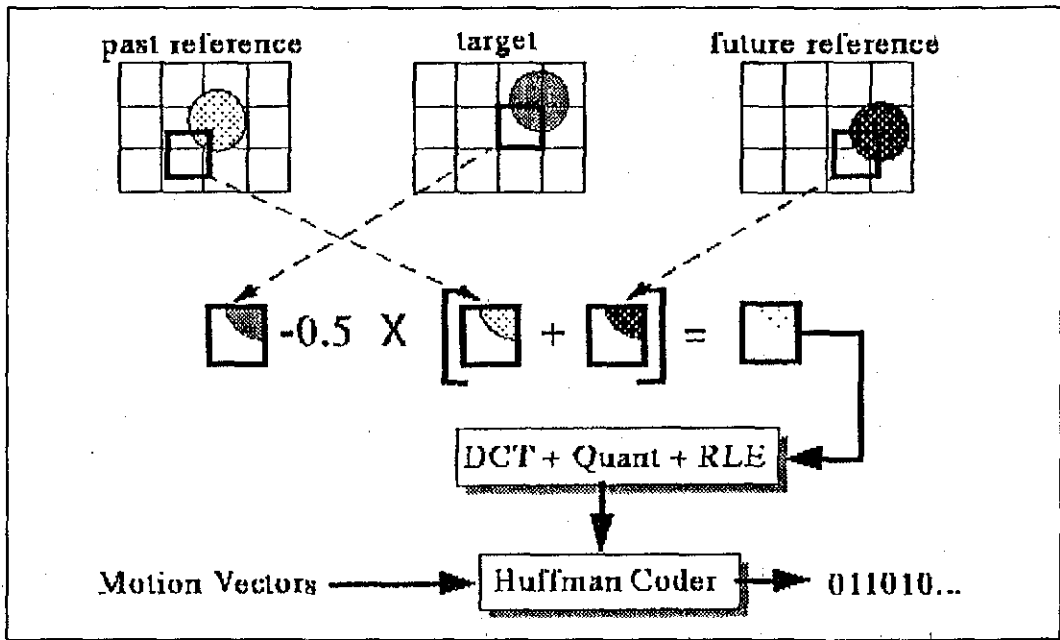


Figure 2.14

2.7.3.1 MPEG File Structure

Figure 2.15 shows the file structure of the MPEG algorithm, which includes the following:

- Sequence Information
 - Seq SC - Sequence Start Code.
 - Video Params, include width, height, aspect ratio of pixels and picture rate.
 - Bitstream Params, are bit rate, buffer size and constrained parameters flag (meaning that the bitstream can be decoded by most hardware decoders).
 - QTs – Quantisation tables, one for I-frames and one for P-frames.
- Group of Pictures (GOP) Information
 - GOP SC – Group Of Pictures Start Code.
 - Time Code, bit field with SMPTE time code (hours, minutes, seconds, frame).
 - GOP Params, are bits describing structure of GOP.
- Picture Information

- PSC – Picture Start Code.
- Type I, P or B-frame.
- Buffer Params, indicate how full the decoder’s buffer should be before starting to decode.
- Encode Params, indicate whether half pixel motion vectors are used.
- Slice Information
 - SSC – Slice Start Code.
 - Vert Pos, what line the slice starts on, or its vertical position within the frame.
 - Qscale, A scaling factor for the quantisation table in the slice.
- Macroblock Information
 - Addr Incr, number of MBs to skip.
 - Type, does this MB use a motion vector? What type?.
 - QScale, A scaling factor for the quantisation table in this MB.
 - Coded Block Pattern (CBP), bitmap indicating which blocks are coded.

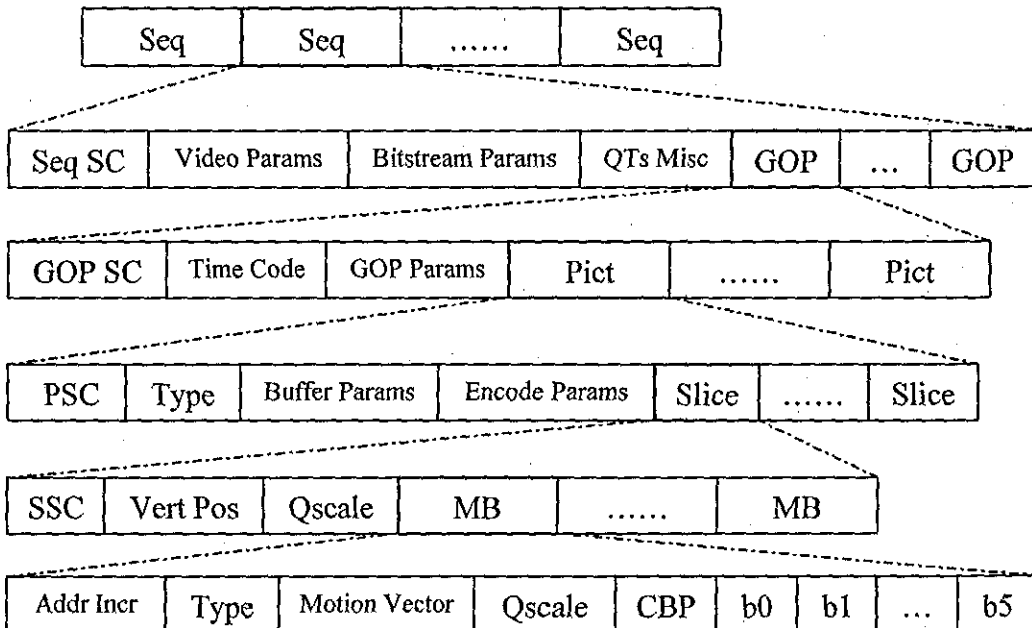


Figure 2.15

2.8 Summary and Conclusions

This Chapter has looked at the stages and various techniques for digitally encoding image and video information. Although not an exhaustive study of all the possible techniques, a brief overview of the related techniques and ideas relevant to this research has been given.

The standards bodies are continually developing the existing standards, for example, the H.263 standard is a backwards compatible update to H.261, where picture quality is greatly improved by using a required $\frac{1}{2}$ pixel motion estimation technique, predicted frames, and a Huffman coding table optimised for low bit rate transmissions. The MPEG standard has many versions e.g. MPEG 1, MPEG 2, MPEG 4. MPEG 1 is designed for the compression of video for storage on CD-ROM and a data rate of 1.5 Mb/s for display on computers [19] i.e. no interlacing required. MPEG 2 is designed for the coding of broadcast video information, where the interlacing of the video information is taken into account.

CHAPTER 3

CHAPTER 3:

TRANSMISSION OF DIGITAL VIDEO ACROSS A COMPUTER NETWORK.

Transmitting digital video across computer networks presents a number of difficulties if the information is to be received correctly, i.e. without errors or lost data, in the correct order or sequence and within a predefined delay. This research has concentrated on the transmission of video data over an ATM based network, attempting to address the problems of cell loss.

Asynchronous Transfer Mode (ATM) is the proposed standard for the Broadband Integrated Services Digital Network (B-ISDN) [2]. ATM was conceived in an attempt to overcome the limitations and inefficiencies of Synchronous Transfer Mode (STM), which is used by telecommunications backbone networks to transfer packetised voice and data across long distances. STM is a circuit switched networking mechanism, where a connection is established between two end points before data is transmitted, and the connection is removed when there is no more data to be communicated. The bandwidth of the STM links is divided into fundamental units of transmission called

time-slots or buckets, which are organised into a fixed number of trains containing a fixed number of buckets, all repeating periodically. On a given STM link, a connection between two end points is assigned a fixed bucket number on a fixed train, and all data to be communicated is carried in this pre-defined bucket. This method is very wasteful of the available bandwidth, especially if the data to be transmitted is bursty in nature. That is, if a connection has periods when no data is transmitted then the corresponding bucket in the pre-defined train goes empty, and can not be used by another connection using the same interconnecting link. Furthermore, the number of possible connections is limited to the total number of buckets on all the different trains.

In addition, the problem of carrying both real time traffic e.g. voice and video, which can tolerate some loss but very little delay, as well as non real time traffic such as computer data and file transfer which can tolerate some delay but no loss, in an integrated way was required. The peak bandwidth requirement of such traffic may be high, as in video data, but the duration of such peaks may be quite small. Therefore, using an STM connection capable of transmitting the peak bandwidth of the traffic is very inefficient. A method where all unused buckets could be used for other connections within a given link would be a better solution.

ATM was thus conceived, where the main idea was that, instead of always identifying a connection by the bucket number, just carry the connection identifier along with the data in any bucket, and keep the size of the bucket small so that if any one bucket is dropped enroute due to congestion, only a small amount of data would be lost. Therefore, two end points in an ATM network are associated with each other via an identifier called the "Virtual Channel Identifier" (VCI) instead of by a time-slot or bucket number as in an STM network. The VCI is carried in the header portion of a packet or cell, the cell being carried in the same type of bucket as per STM, but there is no label or designation for the bucket.

ATM attempts to solve the unused bucket problem of STM by 'statistically multiplexing' several connections on the same link based on their traffic characteristics. That is, if a number of connections on the same link are bursty i.e. peak/average ratio is 10:1 or higher, then they may be assigned to the same link in the

hope that statistically they will not all burst at the same time. If the situation occurs that several of the connections burst simultaneously then there is sufficient elasticity within the network to buffer the burst and to release the data in subsequently available free cells. This allows the situation where the sum of the peak bandwidth requirement of all the connections on a given link can exceed the available bandwidth of the link.

3.1 ATM Cell Structure

As shown in Figure 3.1 an ATM Cell is 53 octets long (1 octet = 8 bits), 5 octets are header information and the remaining 48 are used for data. The 48 octet size represents a compromise between the demand of voice traffic for quick access to the network and the demand of data traffic for large data units. Figure 3.2 shows the structure of the ATM, 5 Octet header.

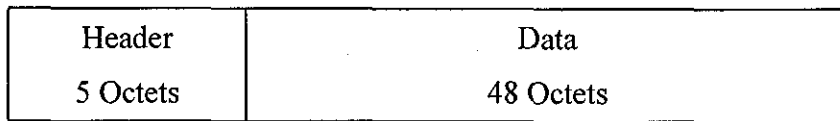


Figure 3.1

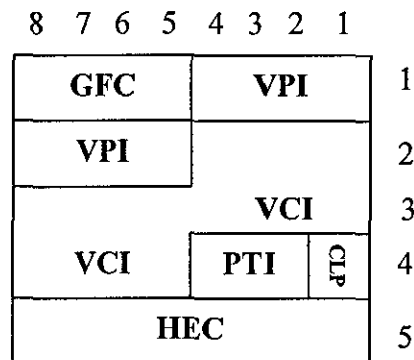


Figure 3.2

The fields in the header have the following uses:

- **GFC** Generic Flow Control, 4 bits used by the flow control mechanism at the User Network Interface (UNI).
- **VPI** Virtual Path Identifier, 8 bits used for directing cells within the ATM network.
- **VCI** Virtual Channel Identifier, 16 bits used for directing cells within the ATM network.
- **PTI** Payload Type Indicator, 3 bits identifies the type of data being carried by the cell.
- **CLP** Cell Loss Priority, 1 bit used to indicate whether a cell is Low Priority and subject to being discarded when the network is congested, or High Priority and less likely to be discarded under congested conditions.
- **HEC** Header Error Correction, 8 bits generated and inserted by the physical layer that serve as a checksum for the first 4 octets of the ATM header. Capable of correcting single bit errors and detecting some multiple bit errors.

The ATM cells flow along virtual channels (VC), which are identified by their virtual channel identifier (VCI). Cells with a common VCI travel exactly the same route within the network and are received in the exact order they were transmitted. VCs are transported within virtual paths (VPs), which are identified by their virtual path identifier (VPI), and are used for aggregating VCs together. The VPI and VCI, taken together form a 24 bit protocol connection identifier (PCI), which identifies a particular call or connection and is used for routing cells across the network and demultiplexing cells at the destination.

3.2 ATM Adaptation Layer

When the network is required to carry data units that are greater than the 48 octets of the ATM cell, an adaptation layer is needed, the ATM adaptation layer (AAL) [20]. The AAL provides for segmentation and reassembly (SAR) of higher layer data units and for detection of errors in transmission. There are a number of AAL protocols each having a certain application area. For example, AAL 1 is meant for constant-bit-rate services such as voice, AAL 2 is meant for variable-bit-rate services with a required

timing relationship between source and destination, such as video. AAL 3 was originally meant for connection-orientated variable-bit-rate services without a required timing relationship, but has now been merged with AAL 4. AAL 5 attempts to reduce the complexity and overhead of AAL 3/4, by removing most of the protocol overhead. AAL 3/4 and AAL 5 contain a convergence sublayer and a SAR sublayer, the AAL 5 SAR sublayer however, is essentially null. The makeup of an AAL 5 convergence sublayer protocol data unit (CS PDU) is shown in Figure 3.3.

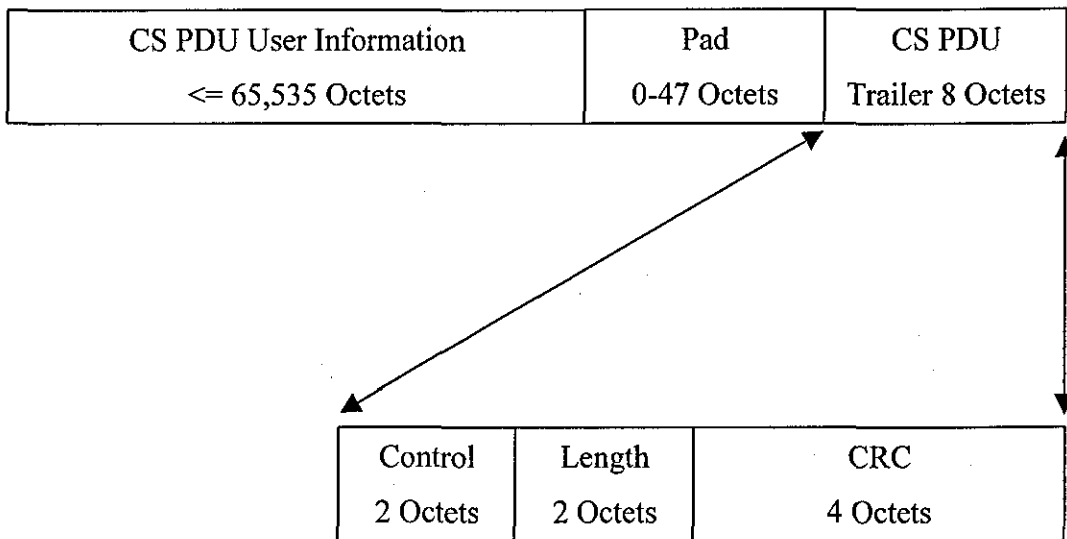


Figure 3.3

The Pad field is used to ensure that the total length of the CS PDU (including the pad and trailer) is divisible by 48 and that the CS PDU trailer is in the last eight octets of the last 48 octet chunk. The use of the Control octets are under study, whilst the Length octets contain the length of the user information in the CS PDU. The CRC, cyclic redundancy check octets, are the check digits of a polynomial error detection code, computed from the bits in the user information in the CS PDU and Pad fields. On reception of the CS PDU, the receiver recomputes the CRC from the CS PDU user information, Pad and CRC fields, and if no errors have occurred then a known result

is obtained, a different result to the expected indicates that an error or bursts of errors have occurred.

When an image or a video is required to be transmitted across an ATM network, the AAL 5 protocol layer first converts the information it is presented with into a CS PDU by adding the pad (if required) and trailer. The CS PDU is then divided into 48 octet SAR PDUs and transmits each in an ATM cell on the same virtual channel. The last SAR PDU is marked so that the receiver can recognise it. Since there is no AAL 5 SAR header, an end of frame indication in the ATM cell header is required. The proposed end of frame indication is an SDU (service data unit) type of 1 in the Payload Type Indicator (PTI) field. On reception, the receiver simply concatenates the cells, watching out for the end of frame indication. When this arrives, the receiver checks the length and CRC and passes the PDU up to the next higher protocol layer. The higher layer is responsible for the actions to be taken on the receipt of CRC errors i.e. whether to discard the entire PDU or attempt to perform some kind of error correction or concealment on the received information.

3.3 ATM Cell Loss

When an ATM network is lightly loaded with traffic, all cells arrive at the receiver in the correct order and within an acceptable delay. However, when there is excessive traffic on the network, either cells may be discarded at intermediate nodes, in an attempt to ease the congestion, or cells may arrive after an unacceptable delay. In addition, cells may also be lost due to random bit errors in the cell header. Therefore, as mentioned above the mechanisms available to compensate for the losses are error correction and error concealment.

3.3.1 Error Correction

Error Correction attempts to recover the lost information exactly, and with regards to the transmission of images or video information, each lost pixel is reproduced as per the original, giving no increase in the Mean Squared Error (or any quality metric) for the received image or frame.

3.3.1.1 Data Retransmission

This is one of the simplest methods of cell recovery, where the information in the lost cell is requested for retransmission. However, such techniques add to network traffic and hence congestion and introduce unacceptable delays in video and voice signals. In addition, when higher layer protocols such as TCP notice that data has been lost, it retransmits the unacknowledged data. The amount of retransmitted data may be considerably more than was discarded, again causing an increase in congestion. Also, the reactive end to end flow control mechanisms of TCP produce further problems when used in a fast (possibly gigabits/second) ATM network. If the flow control is left on an end to end feedback arrangement, then by the time the flow control message is received at the source, several Mbytes of data would have been transmitted into the network, making the congestion even worse. Moreover, by the time the source has reacted to the flow control message, the congestion condition might have disappeared altogether unnecessarily quenching the source.

3.3.1.2 Forward Error Correction

Forward Error Correction Codes add redundant data to the user information, which is used at the receiver to detect and correct a certain number of errors. The two main types of FEC codes are the Bose-Chaudhuri-Hocquenqhem (BCH) codes and the Reed-Solomon (RS) codes. The BCH codes offer flexibility in the choice of code parameters, namely, block length and code rate. BCH codes are among the best known codes of the same size and code rate, at block lengths of a few hundred or less. The RS codes are a subclass of nonbinary BCH codes; the encoder differs from a binary encoder in that it operates on multiple bits rather than individual bits.

The majority of FEC schemes increase the transmission bandwidth to accommodate the coding overheads of the codes, while maintaining a fixed source-coding rate. This in turn has a net negative effect on the reconstructed video quality when the network is heavily loaded, due to an increase in network congestion, which results in higher cell loss rates [3,4]. However, in [21] an attempt is made to reduce the increase in the transmission bandwidth, with a negligible decrease in reconstructed image quality, by implementation of an 'adaptive motion-based FEC coding scheme'.

3.3.2 Error Concealment

Error Concealment is the process of compensating for cell loss using existing information in the data stream. This technique makes use of the visual redundancy in the received images i.e. the lost data is interpolated or deduced from the neighbouring pixels to the error as they are invariably correlated. There are two possible approaches to concealment using video information i.e. Pixel domain techniques and Transform domain techniques.

3.3.2.1 Pixel Domain Error Concealment

Pixel domain error concealment is performed once the image has been reconstructed or decoded back from the compressed state e.g. after variable length decoding, run length decoding, de-quantisation and inverse DCT, for a JPEG encoded image.

Many techniques exist for concealing the lost information in the pixel domain [6, 22, 23], the simplest replace the missing sections of the image by spatial linear interpolation of the surrounding areas. This technique is used when single images are being transmitted i.e. temporally uncorrelated images. When video information is transmitted then both spatial and temporal interpolation can be used, where the algorithm utilises available motion vectors in past reference frames to reconstruct the missing information, in conjunction with pixel data. Other techniques use prioritisation or hierarchical transmission of the compressed data into low priority data such as high frequency coefficients, and high priority data such as low frequency coefficients [6, 24]. This can be achieved by setting the cell loss priority bit (CLP), in an ATM cell header, to zero (Figure 3.2) for high priority cells and to one for low priority cells. Therefore, when congestion occurs the low priority cells are discarded and the high priority cells are retained. However, when congestion increases further the high priority cells may also be discarded, requiring the decoder to perform an error concealment technique as described above.

Techniques that are more complex attempt to recover missing edge information from the edges in the surrounding areas [25, 26, 27, 28]. This is achieved by interpolating

in the direction parallel to the edge or edges. However, the presence of edges in the surrounding image must first be established, along with the position and direction.

3.3.2.2 Transform Domain Error Concealment

The initial methods developed in Loughborough [29, 30, 31], concealed the losses from within the transform domain i.e. after variable length decoding and run length decoding. This approach has the advantage of a much reduced processing load than that required with pixel domain techniques. The loss of a block is detected through the use of markers (JPEG/MJPEG) or start codes (MPEG) that are used to segment the entropy coded data (variable length encoded data). These markers are identifiable within the compressed data, as all markers are reserved bit patterns that do not otherwise occur in the image or video stream, and are byte aligned. The markers are given a predefined sequence that is verified at the receiver. The interval between such markers is not fixed, therefore, the closer the markers the smaller the area of a frame is lost when an error occurs, as resynchronisation is only possible at the marker codes. However, there is a trade off between the size of loss and the amount of redundant information, and hence extra bandwidth, added to the compressed data.

The first technique replaced the lost blocks (MCUs in JPEG and MJPEG, Macroblocks MPEG) with a black or suitable fixed colour block. This method produced very poor results, however, without such an error concealment scheme there is a possibility of a total loss of synchronisation, causing a whole image/frame or a number of frames to be lost. The second technique replaced the lost blocks with blocks from the same position in the previous frame, which gave an improved concealment as long as the frames were correlated with a minimal amount of motion. The third technique replaced the lost blocks with the previous blocks from the same frame, which improved on the second technique when the frames were not correlated e.g. motion or scene change. However, if the error occurs in a region where there is a sharp transition in intensities between the lost and previous blocks e.g. edges, then the results were poor. The fourth technique replaced the loss by linear interpolation of the surrounding blocks. This method produced satisfactory results, as long as the loss did not cross an edge boundary. The processing was performed in the transform domain to take advantage of the energy compaction property of the DCT i.e. the majority of a

blocks energy is concentrated into a small number of transform coefficients, therefore, only a small sub-set of these coefficients are required to be interpolated. This results in a reduced processing load, compared to the amount required for pixel domain interpolation. Further analysis then reduced the amount of interpolated coefficients to four, with a further reduction in processing load, by interpolating the first AC coefficient ($F(0,1)$) from the blocks above and below, the second ($F(1,0)$) from the blocks to the left and right, and the DC coefficient ($F(0,0)$) and fourth AC coefficient ($F(1,1)$) from all the surrounding blocks.

3.3.2.3 Interleaving and Cell Packing Strategies

This technique attempts to prevent the situation where a contiguous number of blocks are lost. If this occurs, the correction/concealment is less effective as the interpolation, or otherwise, is performed on data that is spatially further than the immediately surrounding blocks to the error block. That is, the correlation between any two blocks is proportional to the spatial distance i.e. the greater the separation the smaller the correlation or similarity.

The interleaving of the compressed data can be performed on individual bits, bytes or complete MCUs or MBs [32, 33, 34]. The greater the distance between the neighbouring data, be they bits, bytes e.t.c., the greater the protection against the loss of contiguous data. Interleaving of bits or bytes is preferred when using forward error correction, as these codes are designed to correct a given number of errors within a given number of information bits (BCH) or bytes (RS). Whereas, if an MCU or MB was lost using such codes (depending on their size. See below), the amount of data lost would not be correctable.

In an MPEG sequence, the sizes of the macroblocks can be smaller than the user payload (48 Octets) of an ATM cell, especially in P and B frames. Therefore, multiple macroblocks (MBs) can be packed into a single ATM cell, also, an MCU or MB could be split across a cell boundary, between two cells. Therefore, the transmitter can be made to segment the data in such a way that no macroblock data is split across cell boundaries. Once an integer number of MCUs or MBs is packed into a cell, all the remaining space is padded or filled with zeros, or any predefined pattern of data.

However, there will be occasions where a single MB or MCU will be greater in size than the user payload of an ATM cell. In addition, by the process of inserting padding in certain cells increases the redundant information and hence the data rate or bandwidth, thus increasing the probability of errors in the network.

3.4 Summary and Conclusions

This Chapter has looked at the problems or details of transmitting Digital Video across a computer network, specifically an ATM network. The motivation and structure of an ATM network has also been investigated.

The current methods to compensate for the loss of ATM cells has been investigated, in particular the previous work on the treatment of cell loss at Loughborough University, from which this research has continued.

CHAPTER 4

CHAPTER 4:

TRANSFORM DOMAIN FEATURE DETECTION AND BLOCK CLASSIFICATION.

In order to detect cell losses both Error Correction and Concealment use a technique which by labels are added in some manner to the cells at the transmitter, then the receiver checks the validity of the received sequence of labels. This is the only redundant information introduced when using Error Concealment whereas Error Correction introduces further additional redundant information to perform correction. These labels are called Reset Markers within the JPEG standard (Slices within MPEG) and can be positioned wherever the user requires (subject to certain constraints). The minimum separation between markers is the size of one Minimum Coded Unit (MCU) in JPEG and one Macro Block (MB) in MPEG, i.e. two luminance blocks and two blocks containing chrominance information, (JPEG), four luminance and two chrominance blocks (MPEG). This arrangement ensures that if an MCU/ MB or a portion of an MCU/ MB is missing due to an error only one MCU/ MB is lost and resynchronisation is achieved at the next marker. Obviously if the separation between the markers is increased more MCUs / MBs are lost when an error

occurs. The advantage of increasing the separation is that less redundant information has to be introduced, which reduces the bandwidth required to transmit the image. Once an error has been detected, the first task is to inspect the surrounding blocks to determine whether an edge/feature is present and if it can be used to aid concealment.

4.1 Edge Classification.

The categorising of edges into a number of different groups is a process referred as edge classification. Along with the angle of the edge, its position within the block is also estimated. This information is required within the concealment stage.

The conventional approach to this problem is to use an edge detection algorithm within the pixel domain, which requires a relatively large amount of processing. Kwok and Sun [25] produced an algorithm with an edge classifier that analysed the values of pixels in the blocks surrounding the missing block and determined which edge directions cut through the missing block. However, this required the inspection and processing of a large amount of pixels in each surrounding block. Within this and previous research [31] a method for performing edge classification within the transform domain was investigated to take advantage of the possible reduction in complexity and processing required.

4.2 Initial Classification.

The knowledge of how certain features are represented within the transform domain would aid in the design of a block classification and feature detection algorithm. That is, each feature has a distinctive fingerprint of sorts within the transform domain. First of all a number of black and white blocks with either no edges, vertical, horizontal or diagonal edges (45 degrees) were produced. The DCT operation was then performed on these blocks to analyse the relative amplitudes of all the coefficients within each block. Figure 4.1 shows the ten images produced for the ten chosen classes.

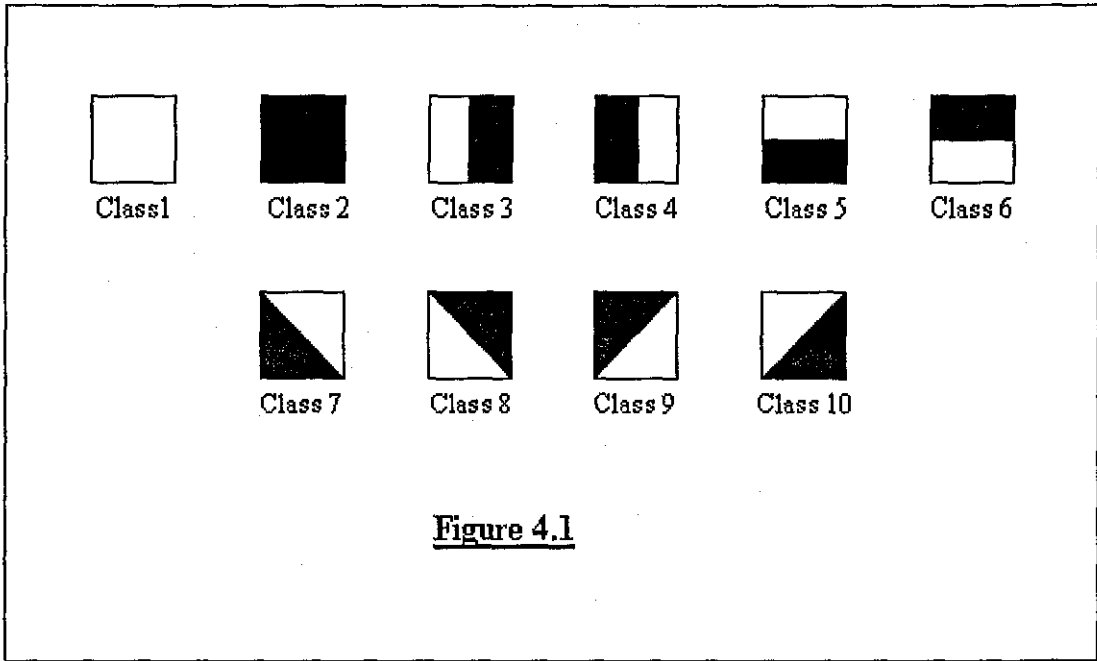


Figure 4.1

The first two images (class1&2) are blocks with no edge information i.e. the luminance values are virtually constant throughout the entire block. This class of block is identified by a dominant DC value, coefficient $F(0,0)$ within the transform domain. A positive DC value would give a class 1 block and a negative DC value would give a class 2 block. The next two images (class 3&4) are blocks with a vertical edge in the centre of the block composed of a vertical strip of 4 white/black pixels followed by a strip of 4 black/white pixels. With a white pixel having a value of +127 (after subtraction of 128), and a black pixel having a value of -128 (after subtraction of 128).

$$\begin{bmatrix}
 -128 & -128 & -128 & -128 & -128 & -128 & -128 & -128 \\
 127 & -128 & -128 & -128 & -128 & -128 & -128 & -128 \\
 127 & 127 & -128 & -128 & -128 & -128 & -128 & -128 \\
 127 & 127 & 127 & -128 & -128 & -128 & -128 & -128 \\
 127 & 127 & 127 & 127 & -128 & -128 & -128 & -128 \\
 127 & 127 & 127 & 127 & 127 & -128 & -128 & -128 \\
 127 & 127 & 127 & 127 & 127 & 127 & -128 & -128 \\
 127 & 127 & 127 & 127 & 127 & 127 & 127 & -128
 \end{bmatrix}
 \Leftrightarrow
 \begin{bmatrix}
 -131 & 581 & 0 & 61 & 0 & 18 & 0 & 5 \\
 -580 & -127 & 266 & 0 & 48 & 0 & 15 & 0 \\
 0 & -265 & -127 & 151 & 0 & 30 & 0 & 7 \\
 -60 & 0 & -150 & -127 & 98 & 0 & 19 & 0 \\
 0 & -47 & 0 & -97 & -127 & 65 & 0 & 10 \\
 -17 & 0 & -29 & 0 & -64 & -127 & 42 & 0 \\
 0 & -14 & 0 & -18 & 0 & -41 & -127 & 22 \\
 -4 & 0 & -6 & 0 & -9 & 0 & -21 & -127
 \end{bmatrix}$$

Figure 4.2.1

After a DCT transformation, the class 3 block produces the following coefficient values: $F(0,0) = -3$, $F(0,1) = 924$, $F(0,2) = 0$, $F(0,3) = -324$, $F(0,4) = 0$, $F(0,5) = 217$, $F(0,6) = 0$, $F(0,7) = -183$, $F(1 \text{ to } 7, 0 \text{ to } 7) = 0$. As can be seen in Figure 4.2.1, there are values on the first row only all other rows are zero. The luminance values model a square wave characteristic in the x-axis and the Fourier Series of a 'periodic' square wave gives a fundamental and all the odd harmonics with reducing magnitude ($1/3, 1/5, 1/7$, e.t.c.). It is therefore not surprising that the DCT coefficients resemble this result as the DCT is closely related to the DFT, which takes the eight samples and makes a 'periodic' function of the samples to give discrete frequency values. The coefficient values of class 4 blocks are the same as those for class 3 but negated (opposite in sign) as the luminance values are the negative of those for class 3 blocks. The coefficients for class 5&6 are the same as those for class 3&4 respectively, but are found in the first column instead of the first row. This is as expected as the edge in the luminance block has been rotated by 90 degrees and the basis images of the DCT have this relation. That is the basis images of the first column are rotated versions of the basis images of the first row (90 degree rotation), this applies to all rows and columns of the 2-dimensional matrix of DCT basis images, an orthogonal transform. Therefore, the transform matrix of a class 5/6 block is the transpose of a class 3/4 block.

The images in class 7&8 are blocks with a diagonal edge at 45 degrees from left to right. The following two matrices in Figure 4.2.1 show a pixel domain matrix of a class 7 block on the left and the resulting transform domain representation after a DCT transformation on the right. The transform coefficient matrix in Figure 4.2.1 has anti-symmetry along the leading diagonal and the most significant values are in those coefficients closest to this diagonal, coefficients $F(0,1)$, $F(1,0)$, $F(1,1)$, $F(2,1)$, $F(1,2)$, $F(2,2)$ having noticeable prominence. The coefficients for a class 8 block would have nearly exactly the same values but negated as would be expected, as this class is the negative of a class 7 block. Class 9&10 are blocks with the diagonal edge at 45 degrees from right to left, or class 7&8 blocks rotated by 90 degrees.

$$\begin{bmatrix}
 -128 & -128 & -128 & -128 & -128 & -128 & -128 & -128 \\
 127 & -128 & -128 & -128 & -128 & -128 & -128 & -128 \\
 127 & 127 & -128 & -128 & -128 & -128 & -128 & -128 \\
 127 & 127 & 127 & -128 & -128 & -128 & -128 & -128 \\
 127 & 127 & 127 & 127 & -128 & -128 & -128 & -128 \\
 127 & 127 & 127 & 127 & 127 & -128 & -128 & -128 \\
 127 & 127 & 127 & 127 & 127 & 127 & -128 & -128 \\
 127 & 127 & 127 & 127 & 127 & 127 & 127 & -128
 \end{bmatrix}
 \Leftrightarrow
 \begin{bmatrix}
 -131 & 581 & 0 & 61 & 0 & 18 & 0 & 5 \\
 -580 & -127 & 266 & 0 & 48 & 0 & 15 & 0 \\
 0 & -265 & -127 & 151 & 0 & 30 & 0 & 7 \\
 -60 & 0 & -150 & -127 & 98 & 0 & 19 & 0 \\
 0 & -47 & 0 & -97 & -127 & 65 & 0 & 10 \\
 -17 & 0 & -29 & 0 & -64 & -127 & 42 & 0 \\
 0 & -14 & 0 & -18 & 0 & -41 & -127 & 22 \\
 -4 & 0 & -6 & 0 & -9 & 0 & -21 & -127
 \end{bmatrix}$$

Figure 4.2.2

The transform domain coefficients are the same as those for class 7&8 blocks but the transform matrix is symmetric instead of anti-symmetric i.e. the values either side of the leading diagonal have the same sign.

In the JPEG and MPEG standards once the DCT transformation has been completed the coefficients are quantised to remove most of the high frequency coefficients to facilitate compression. The Luminance Quantisation table for the JPEG standard is shown in Figure 4 2.3.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 4.2.3

As can be seen emphasis is given to the top left hand corner of the matrix in Figure 4.2.3 i.e. the first 15 coefficients in a zigzag scan from coefficient $F(0,0)$. When the transform coefficient matrices for classes 3 to 10 are quantised most of the high frequency information is lost and approximately, only the first 15 coefficients are significant. For classes 3, 4, 5 and 6 this implies that only the fundamental, ($F(0,1)$ or $F(1,0)$) and the third harmonic ($F(0,3)$ or $F(3,0)$) will have values that are significant. For classes 7, 8, 9 and 10 the significant coefficients are: $F(0,1)$, $F(1,0)$, $F(1,1)$, $F(1,2)$, $F(2,1)$, $F(2,2)$. Therefore, as expected each class or edge/feature has a distinguishable fingerprint within the transform domain which can be used for block classification, however due to the fact that quantisation removes the high frequency component of this fingerprint only the low frequency coefficients are of any use.

The first and most intuitively obvious approach to classification of the blocks is to look at the magnitude of the first few transform domain coefficients as identified above, to establish if there is an edge present within the block under observation. This approach was used in previous research [31] where a simplified classification method concentrated on the two 'low frequency' AC coefficients $F(0,1)$ and $F(1,0)$ only. The relative values of the two coefficients were firstly compared. A strong component in AC coefficient $F(0,1)$ 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 edge within the block. If both coefficients were greater than a pre-defined value and the two coefficients were similar, a diagonal edge within the block would be indicated. If the block failed the above tests and both $F(0,1)$ and $F(1,0)$ were below another pre-defined value the block would be classified as a flat block otherwise it would be classified as an unknown block. Blocks having multiple edges or fine texture are some of the examples of blocks, which would produce a classification of unknown. Therefore, the classifier used five classes i.e. horizontal, vertical, diagonal, flat and unknown as shown in Figure 4.3. However, this approach of analysing the values of coefficients is restrictive and limits the detectable features or edges to those having a large difference in contrast between either side of the edge, some edges are incorrectly classified and others are missed completely. A probable contributing factor to these failings is the fact that '*Weber's Law*' [35] is not taken into account when the pre-defined thresholds were calculated. '*Weber's Law*' is explained in section 4.3 following.

In addition, the algorithm does not assign an angle to the edge nor its position within a block. This severely limits the performance of the concealment stage when an edge of angle other than 45 degrees is encountered. Also, there are occasions when the edge in the surrounding block does not enter the error block due to its angle and position, but the *Initial Classifier* in conjunction with a *Concealment* stage is incapable of detecting this. To remove these failings a new method of edge detection and classification within the transform domain was developed.

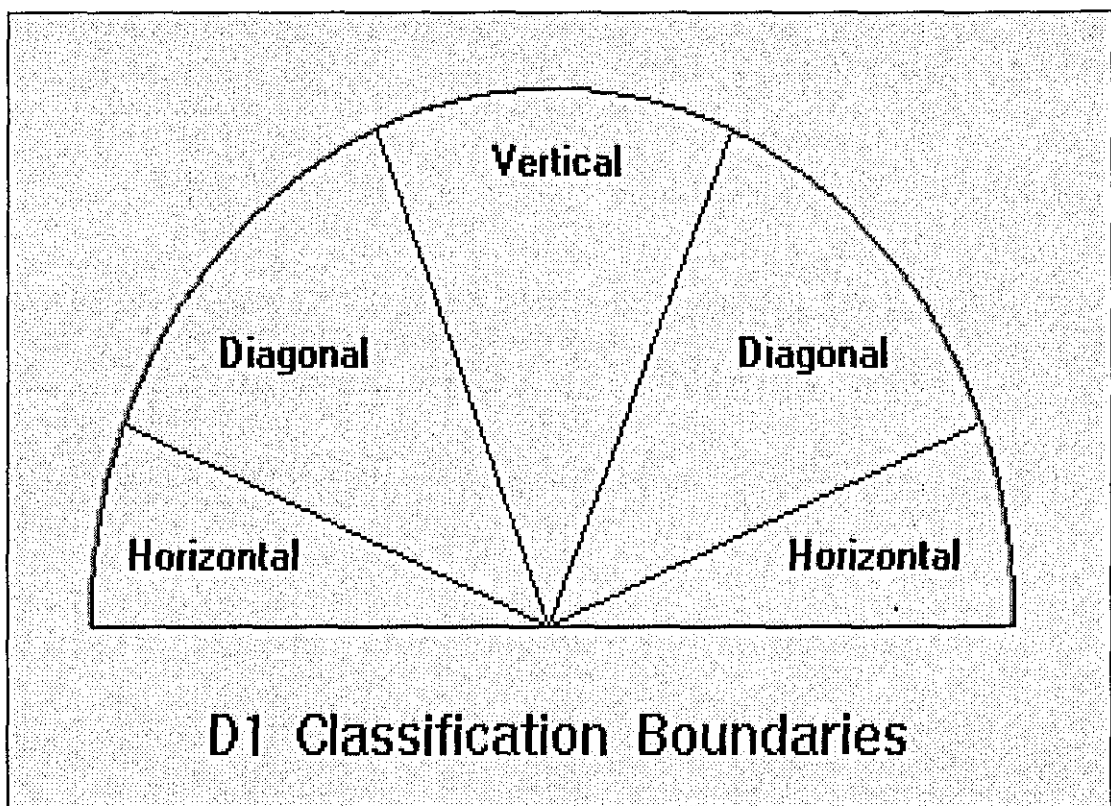


Figure 4.3

4.3 Advanced Classification.

After extensive experimentation, it was found that the ratios of certain coefficients could be used to indicate the presence of edges within a block. These ratios are referred to as R1, R2 and R3 in the following sections. Specifically, only three ratios

are calculated and due to the choice of coefficients, they can be used to indicate the position/displacement and angle of an edge irrespective of the contrast on either side of the edge. That is for a given angle and displacement of a single edge the values of certain ratios are practically identical when the contrast between either side of the edge is varied as shown in Figure 4.4/5/6. This is the case when the difference in pixel values on either side of the edge is as small as 5, and when the peak values are -128 and +127. Obviously, the human eye cannot detect such a small change, therefore there is a threshold below which the edge is not visible to the human eye. This will not be a fixed threshold due to 'Weber's Law', which states that the just noticeable difference between two adjacent regions of pixels is proportional to the Intensity. As the Intensity increases, a larger difference between the two regions is required to make one region noticeably different from the other.

As mentioned earlier, in the JPEG and other standards, once the DCT transformation has been completed the coefficients are quantised to remove most of the high frequency coefficients and hence facilitate compression. The detection and classification process therefore uses a combination of three of the remaining low frequency coefficients, which were shown significant in the previous section 4.2. The significant coefficients when an edge is present in an 8x8 block are AC01, AC10, AC11, AC12, AC21, and AC22.

Now: -

$$R1(n) = AC\ 01(n) / AC\ 10(n). \quad \dots\dots (5.0.1)$$

$$R2(n) = AC\ 11(n) / AC\ 01(n). \quad \dots\dots (5.0.2)$$

$$R3(n) = AC\ 11(n) / AC\ 10(n). \quad \dots\dots (5.0.3)$$

Where n is the value of an edges position/displacement.

Figure 4.4
Ratio Information for diagonal edges at a fixed position/displacement n, and varying Contrast either side of edge.

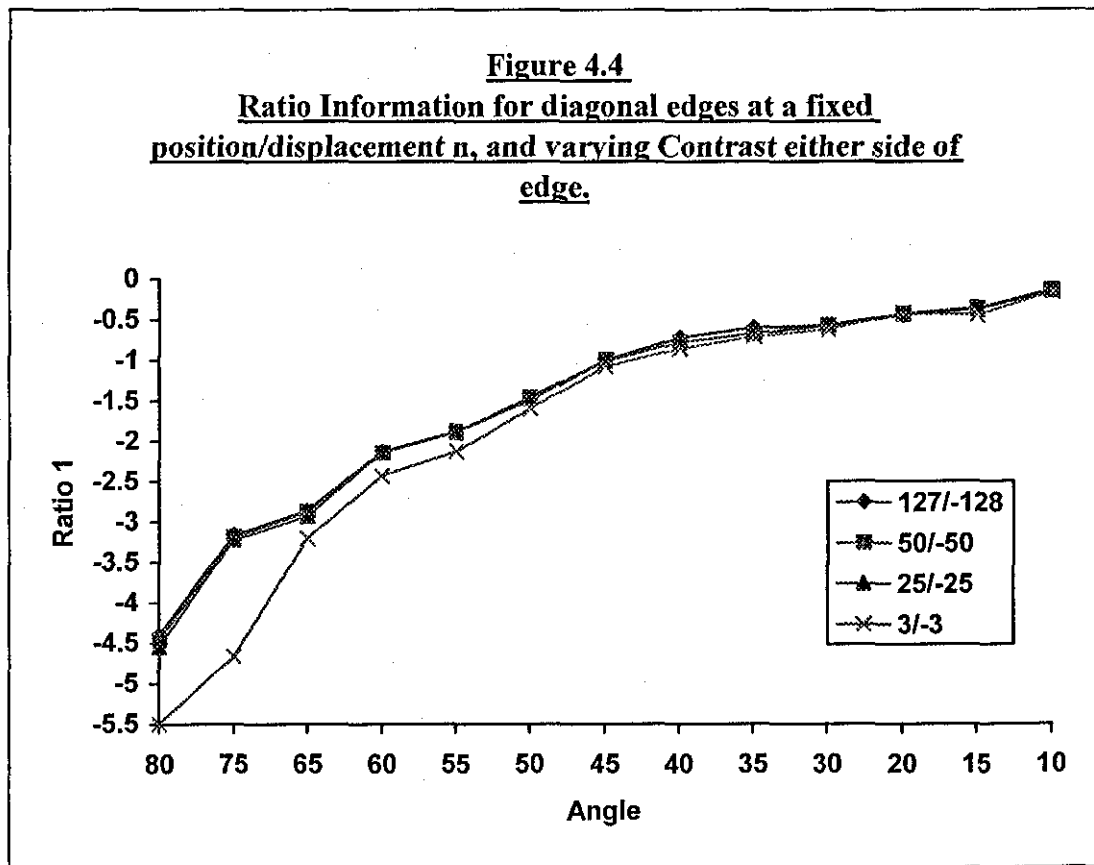
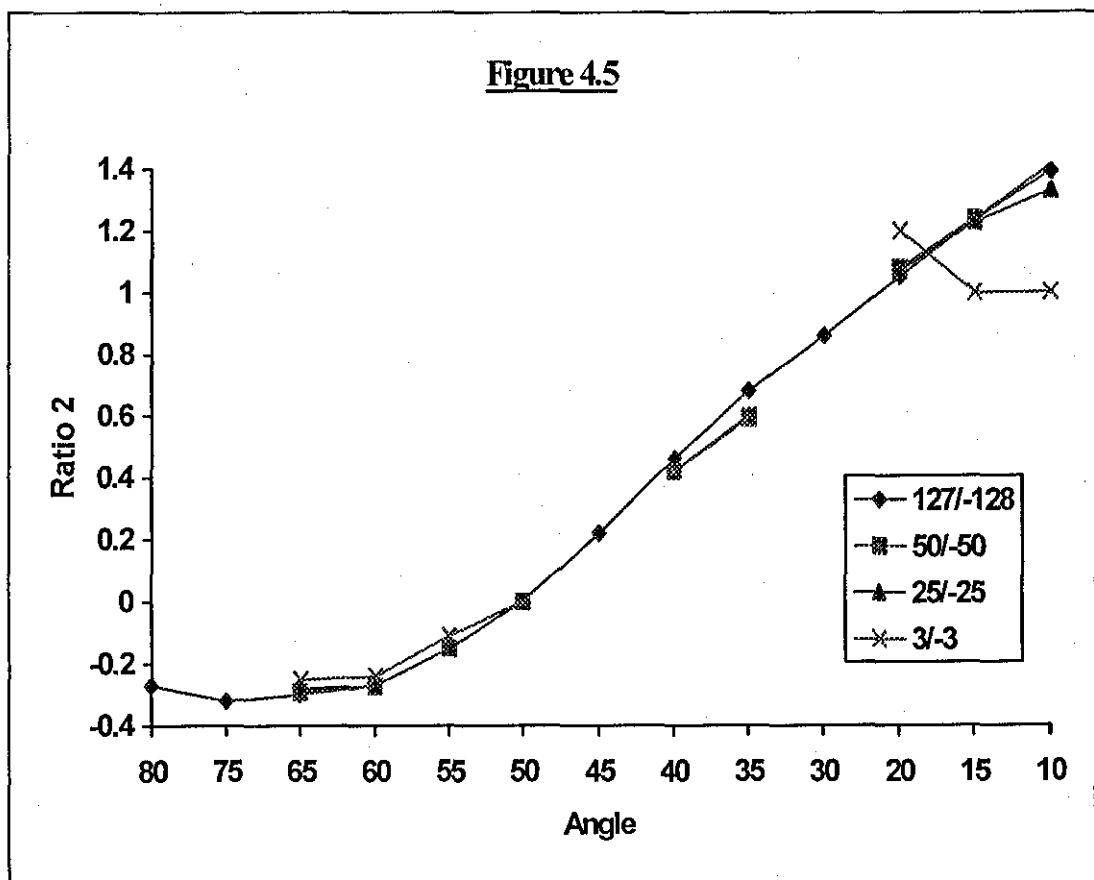
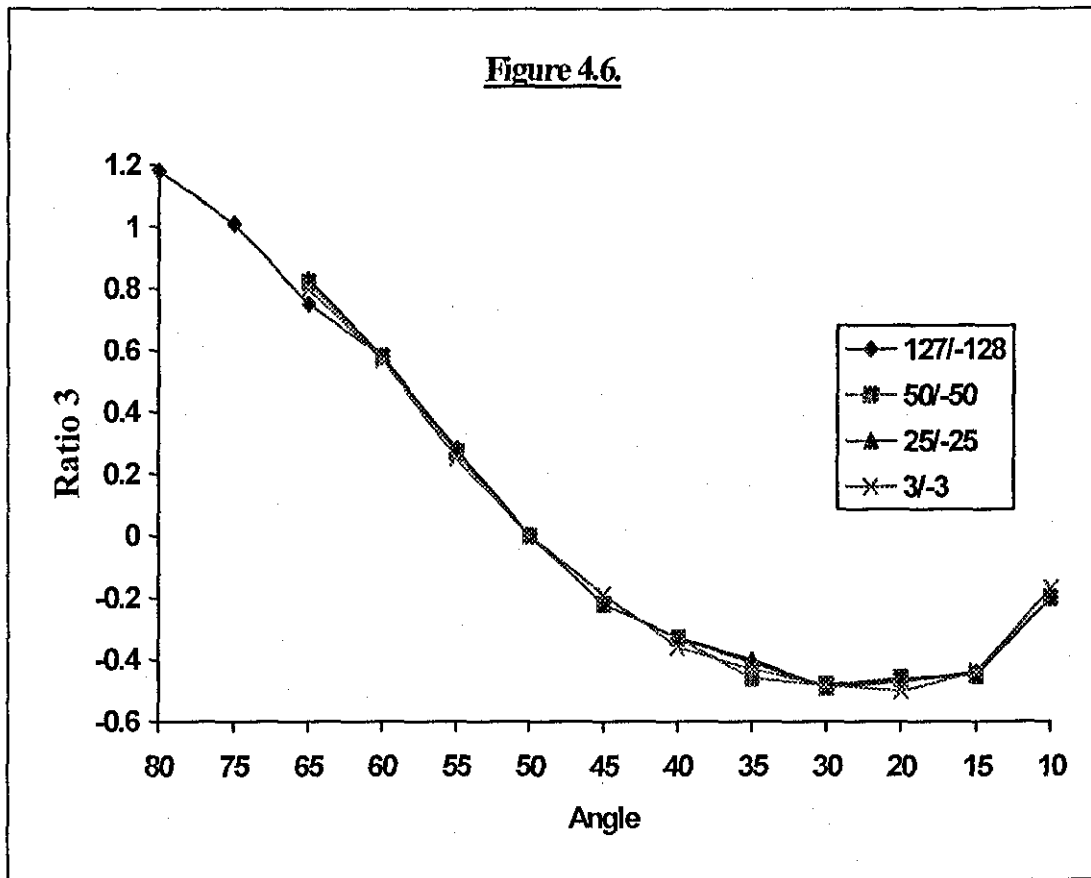


Figure 4.5





The graphs in Figures 4.4 to 4.6 show the value of Ratio 1, 2 and 3 for various angled edges having the same value of displacement or position n . That is, the edges all originate from the same pixel in an 8 by 8 pixel block, pixel (0,0) being defined as the top left pixel of the block. The angles in the graphs in Figures 4.4 to 4.6 are produced when the individual edges pass through one of the pixels in the bottom row or rightmost column of the 8x8 block. An example of such a block is shown in Figure 4.7.

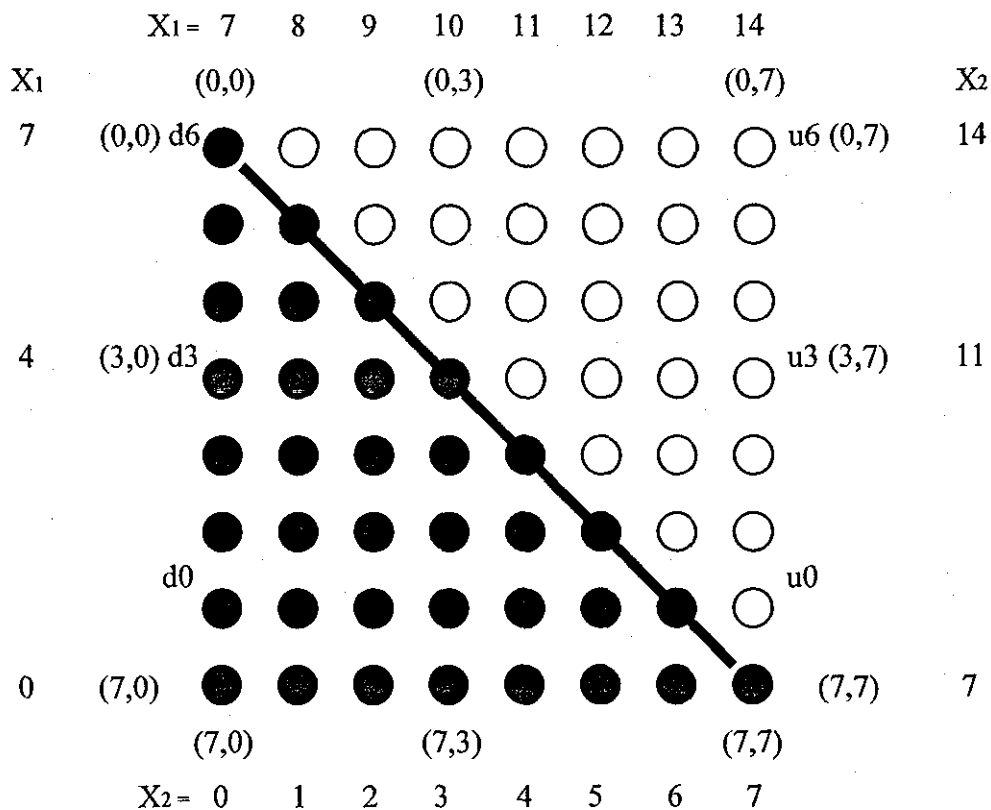


Figure 4.7

The above block in Figure 4.7 consists of 8 by 8 circles representing 8 by 8 pixels, with some of the pixels having been labelled with their respective co-ordinates. The leading diagonal has been represented by a black line and the pixels in the top half to the right of the leading diagonal are white representing a value of +127 (Luminance value), the pixels to the left of the leading diagonal are black representing a Luminance value of -128. The edge in the block represented by the black line has an angle of 45 degrees and originates from the top left pixel i.e. pixel (0,0), this is assigned a value of 6 for the displacement of the edge, represented by the d_6 next to the pixel. An edge with a displacement of 0 originates from pixel (6,0), as a down hill edge originating from pixel (7,0) would not enter the block in question. When an edge has a displacement greater than 6 the edge originates from a pixel in the first column but from the block above. However, the edge begins from the top row of pixels in the block. The edges that begin at the top left and end at the bottom right are assigned as 'down hill' edges. If the edge begins at the top right and ends in the bottom left it is assigned as an 'up hill' edge. In up hill edges the assignment of displacement is the

same except that the edges originate from pixels in the last column of the block instead of the first, and end or pass through the bottom row or first column of the block. The prefixes d and u before the value of displacement in the diagram in Figure 4.7 indicates an up hill or down hill value of displacement. If the bottom row and right most column of pixels are numbered $X_2 = 0$ to 14 then the edges in the graph of Figure 4.7 can be assigned this value, instead of the edges angle. Also if each pixel in the first column and first row are numbered $X_1 = 0$ to 14, and a set of edges originating from each value of X_1 and ending at each value of X_2 a three dimensional graph would result, showing the Ratio value for the start point and end point within the block of pixels. Figures 4.8, 4.9 and 4.10 show these graphs for ratios R1, R2 and R3. For certain values of X_1 and X_2 the values of R1, R2 and R3 go off the scale. This occurs when the edge is either vertical (R1 and R3) or horizontal (R2), due to the fact that coefficient AC10 or AC01 is zero when vertical or horizontal edges are classified. This information can be used to detect vertical or horizontal edges. Also there is a simple test to see if an edge is up or down hill, for down hill edges AC01 and AC10 are opposite in sign, where as in up hill edges AC01 and AC10 have the same sign.

As can be seen from the characteristics in Figure 4.8, each line representing a particular position of X_1 has a unique value for each value of X_2 , that is given a value for each Ratio 1, 2 and 3 there will only be one value for X_1 and X_2 , i.e. uniquely decodable. This value for X_1 and X_2 could be found using Ratios R2 and R3 alone.

Figure 4.8

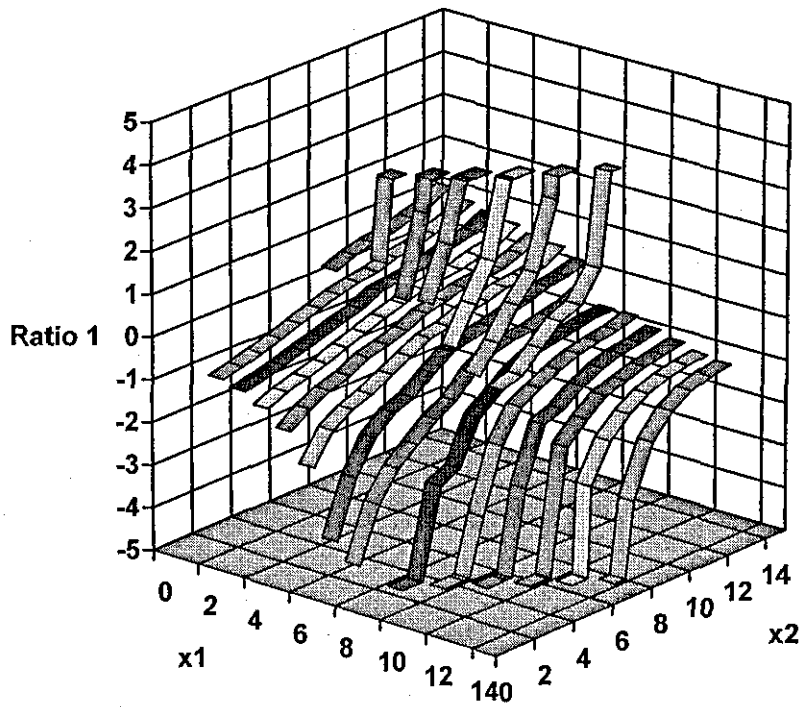


Figure 4.9

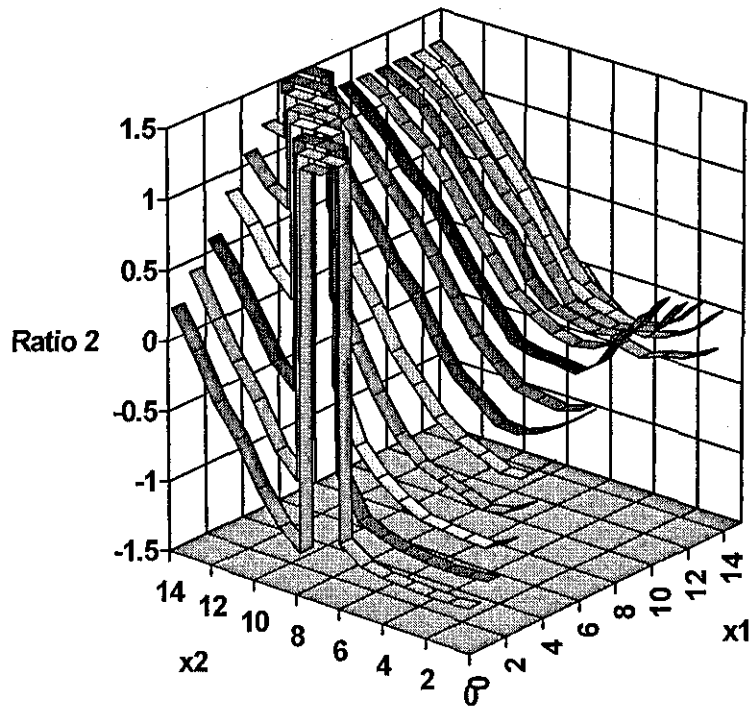
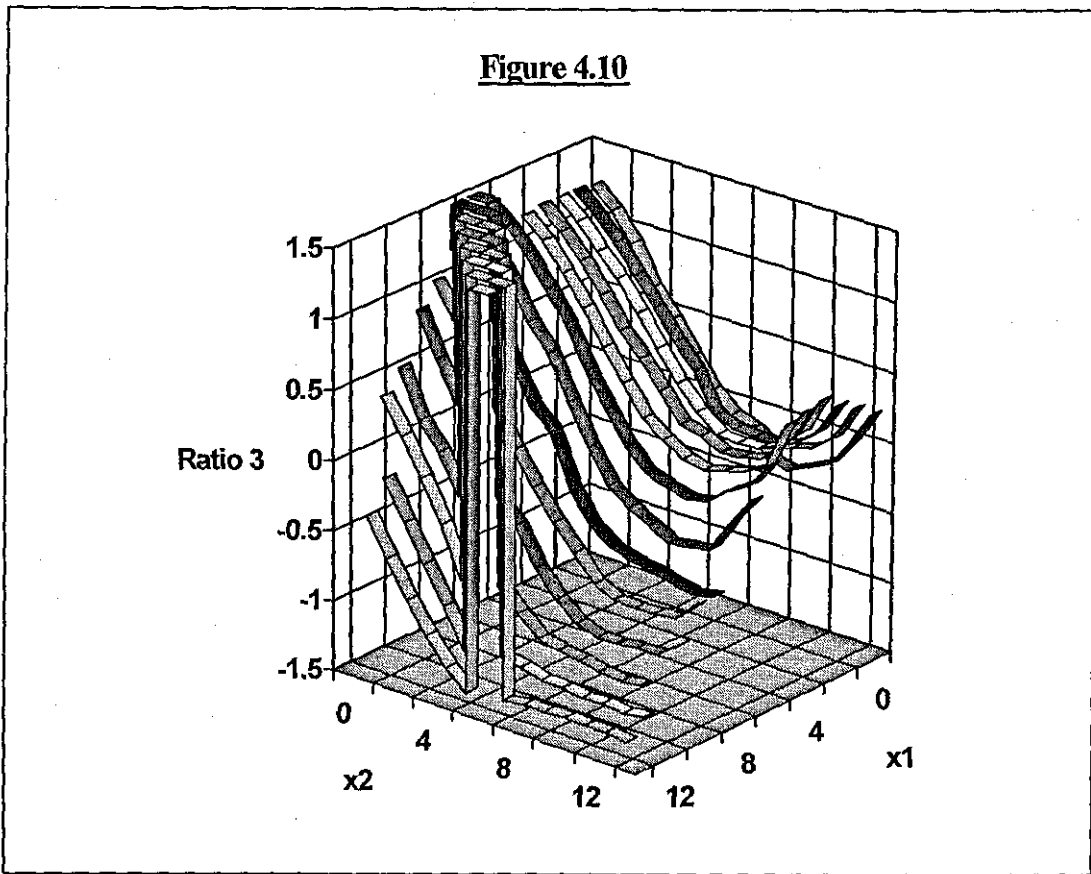


Figure 4.10

The graphs in Figures 4.8, 4.9 and 4.10 include all the downhill edges possible, but some of the possible up hill edges are not included. Also due to the relatively small size of the pixel blocks used within JPEG, MJPEG and MPEG and the tendency for most edges to maintain their angle [36] i.e. straight, especially within the space of three 8x8 pixel blocks, the edge within the error block will be a displaced version of the edges in the surrounding blocks. For these reasons, graphs of Angle against Displacement were used in the classification process, as shown in Figures 4.11, 4.12, 4.13, 4.14, 4.15, 4.16 and 4.17. Coefficients AC12, AC21 and AC22 were identified in section 4.1 as being significant when classifying single edges, however these coefficients were not used in the classification algorithm. The ratios produced using these coefficients aren't suited to the classification process, giving a number of possible answers for a given ratio value or combination of ratios. Also the ratios produced can not be modelled or represented empirically by simple equations, as is the case with the ratios R1, R2 and R3. Some examples of the possible ratios produced using these unused coefficients are given in figures 4.18, 4.19, 4.20, 4.21 and 4.22.

Figure 4.11

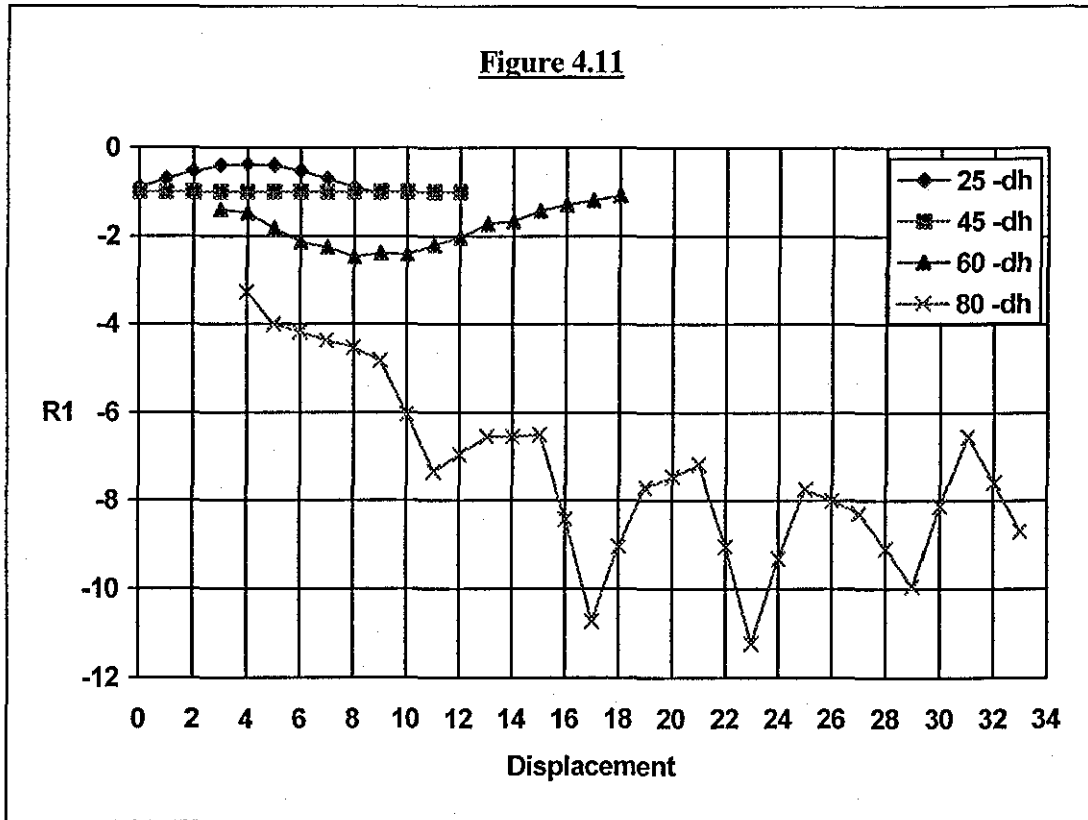


Figure 4.12

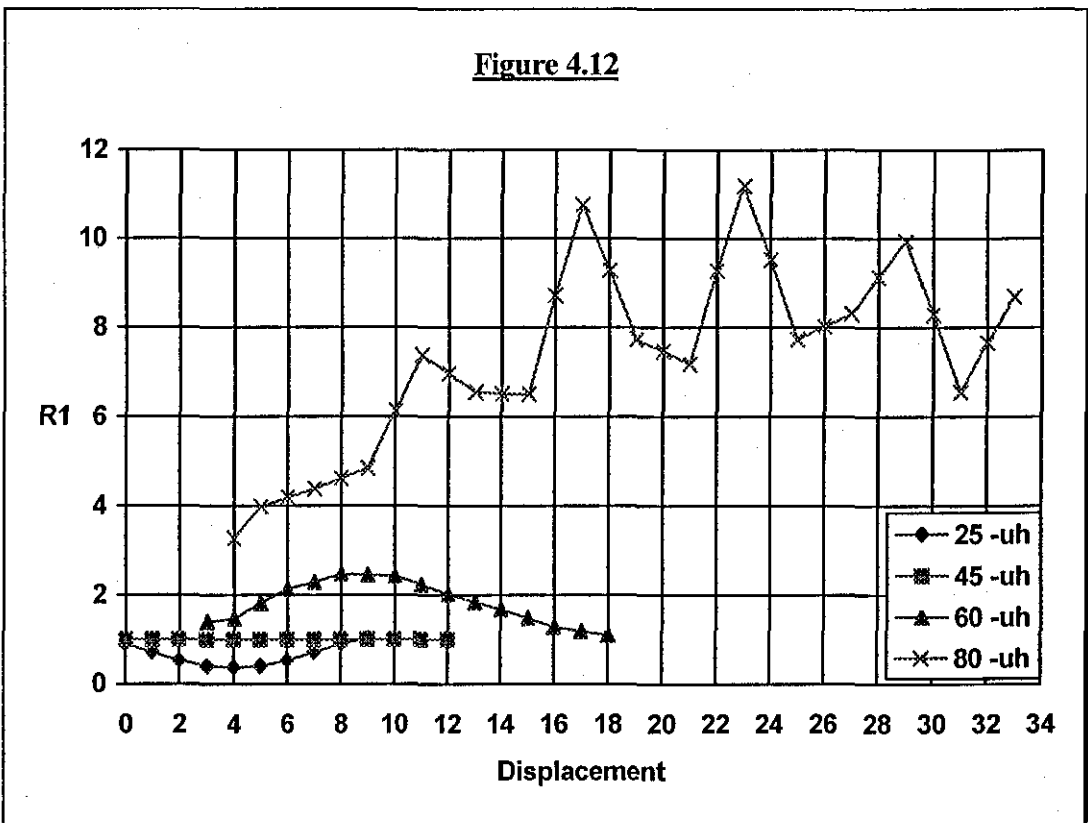


Figure 4.13

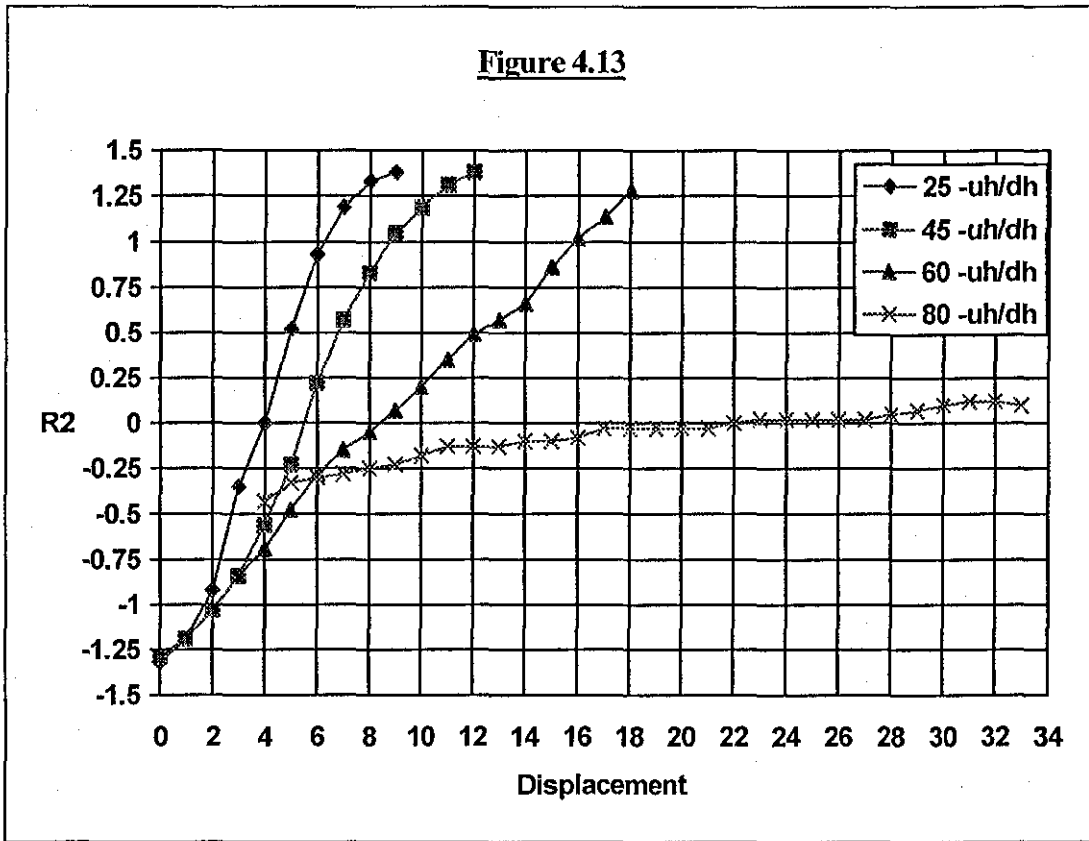


Figure 4.14

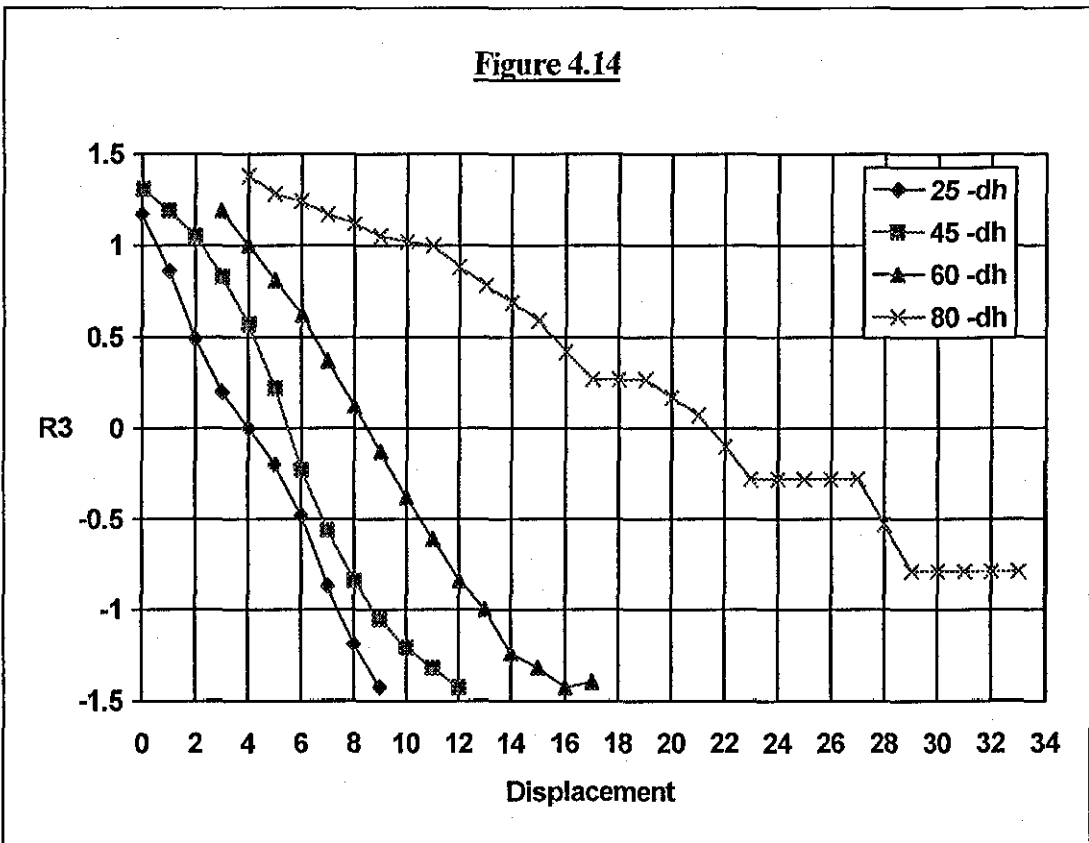


Figure 4.15

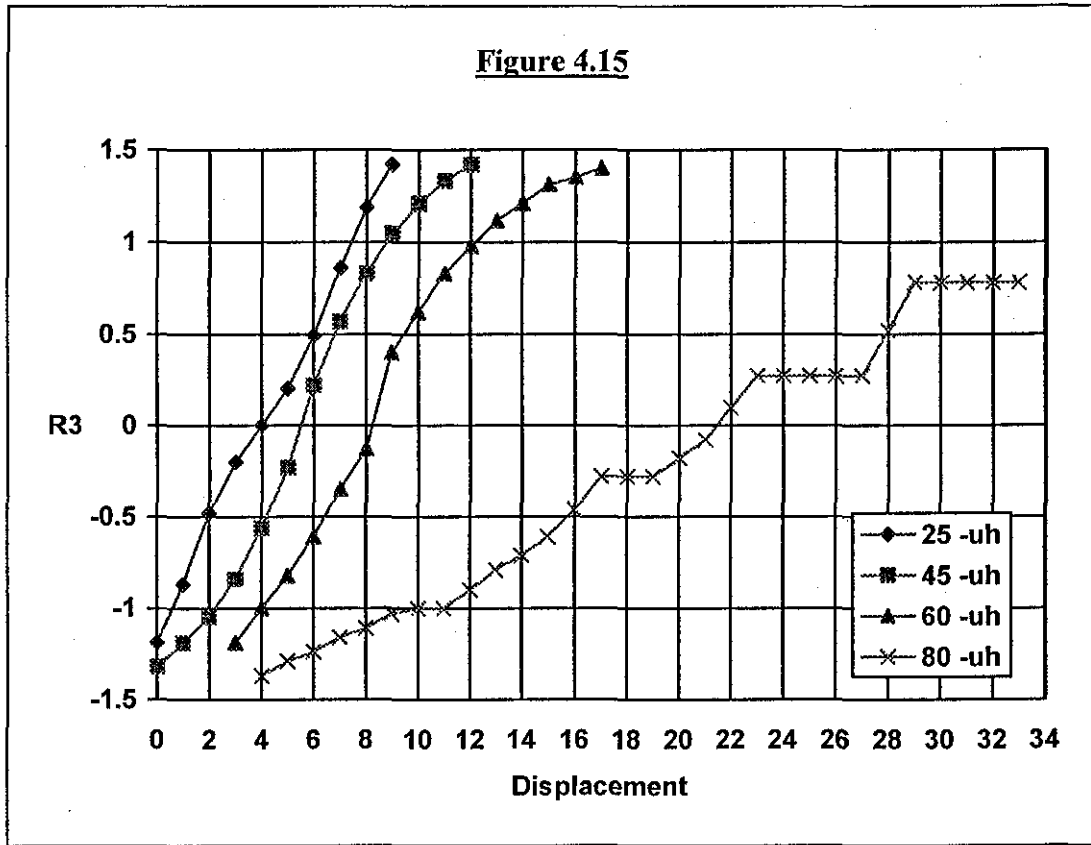
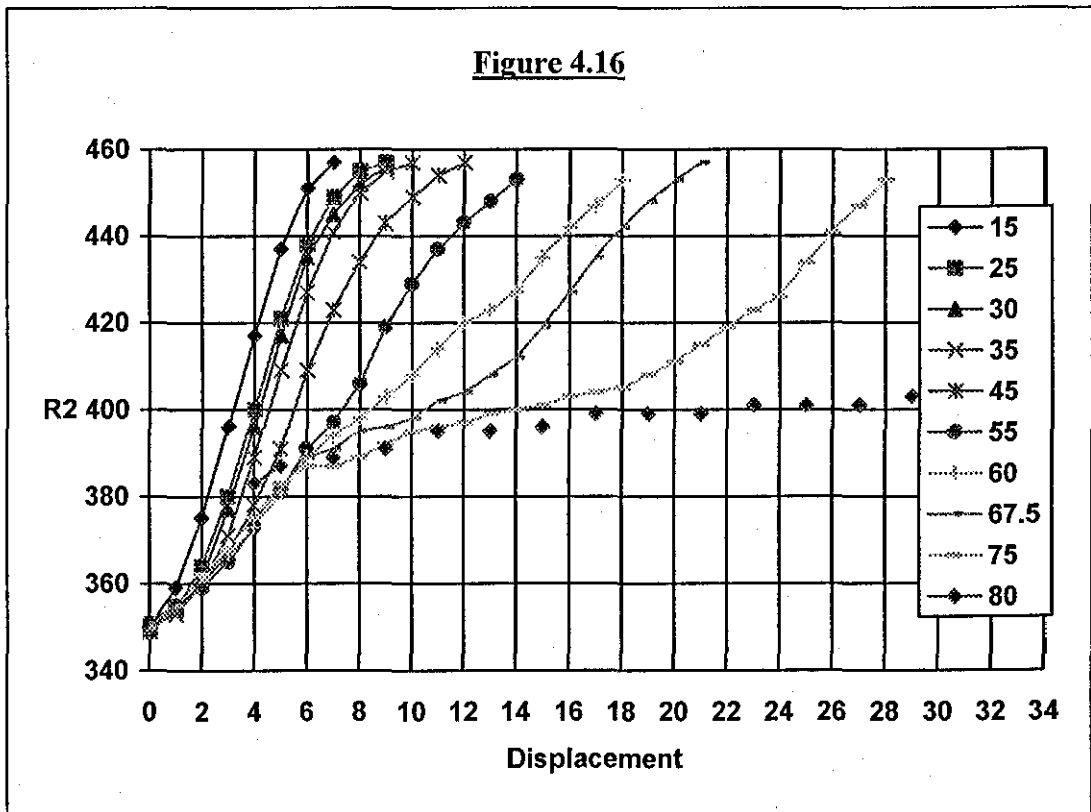


Figure 4.16



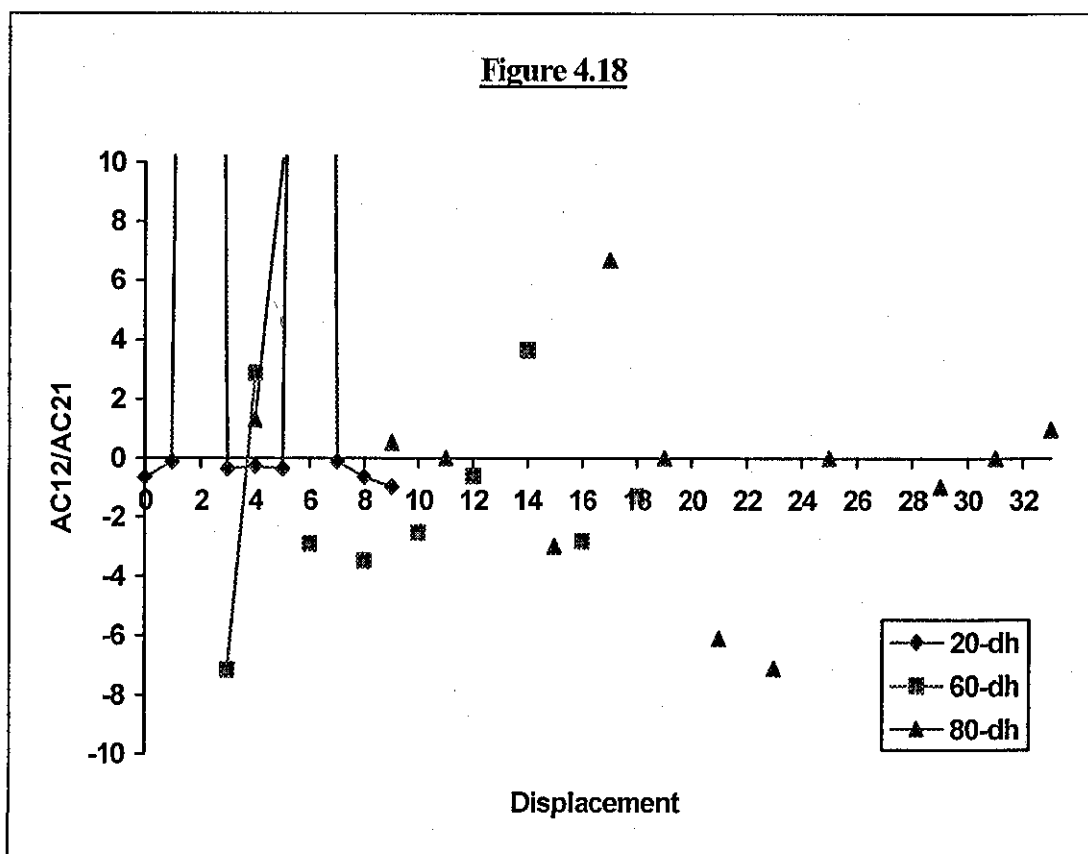
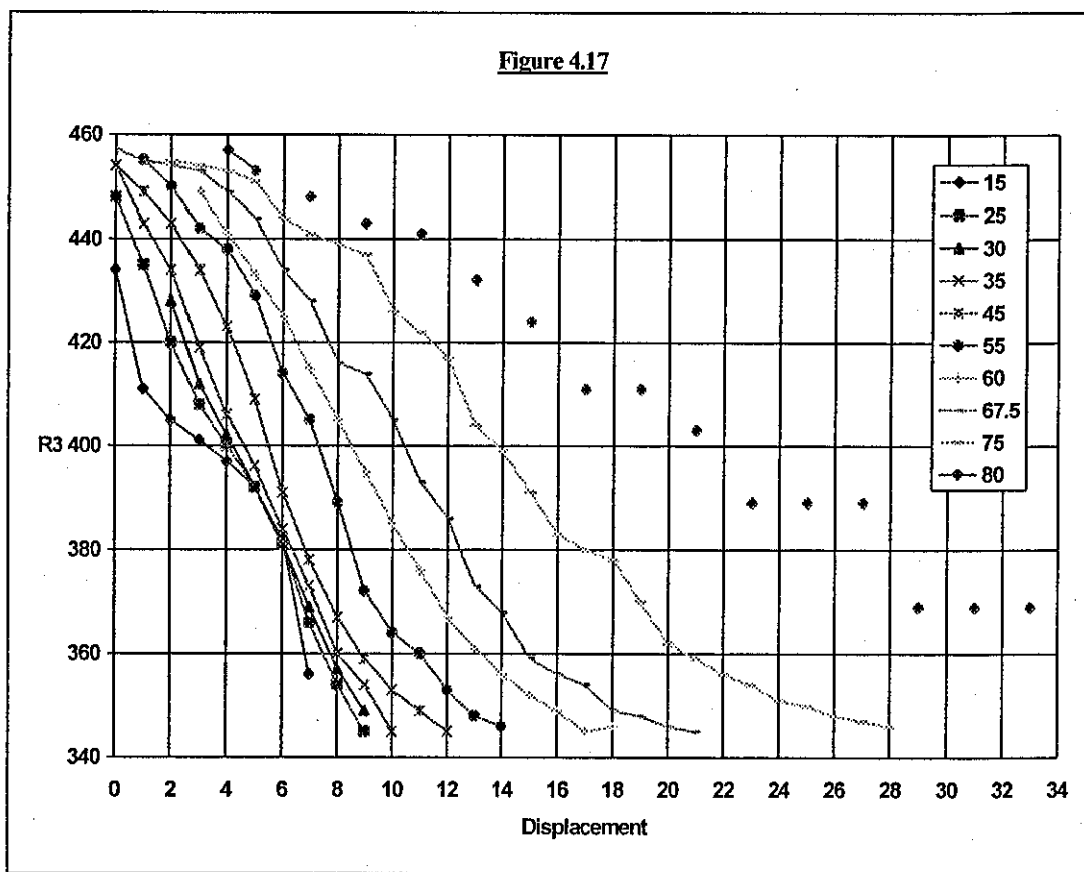


Figure 4.19

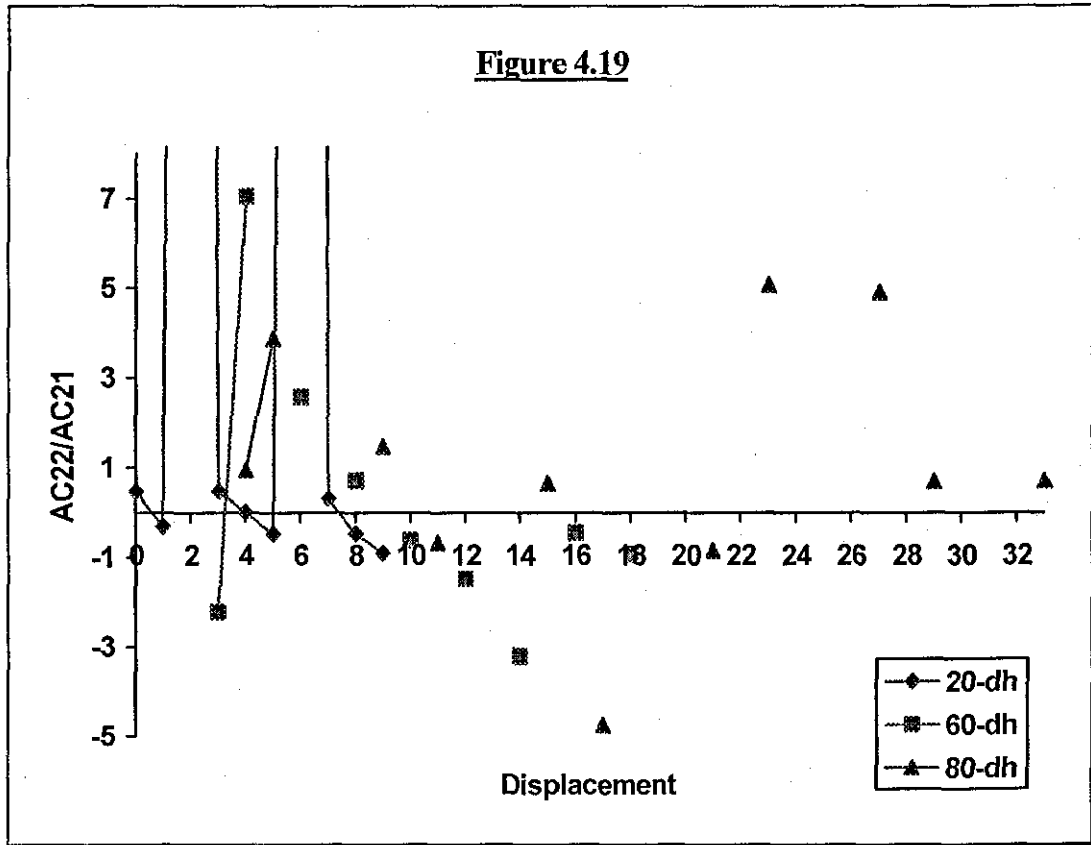


Figure 4.20

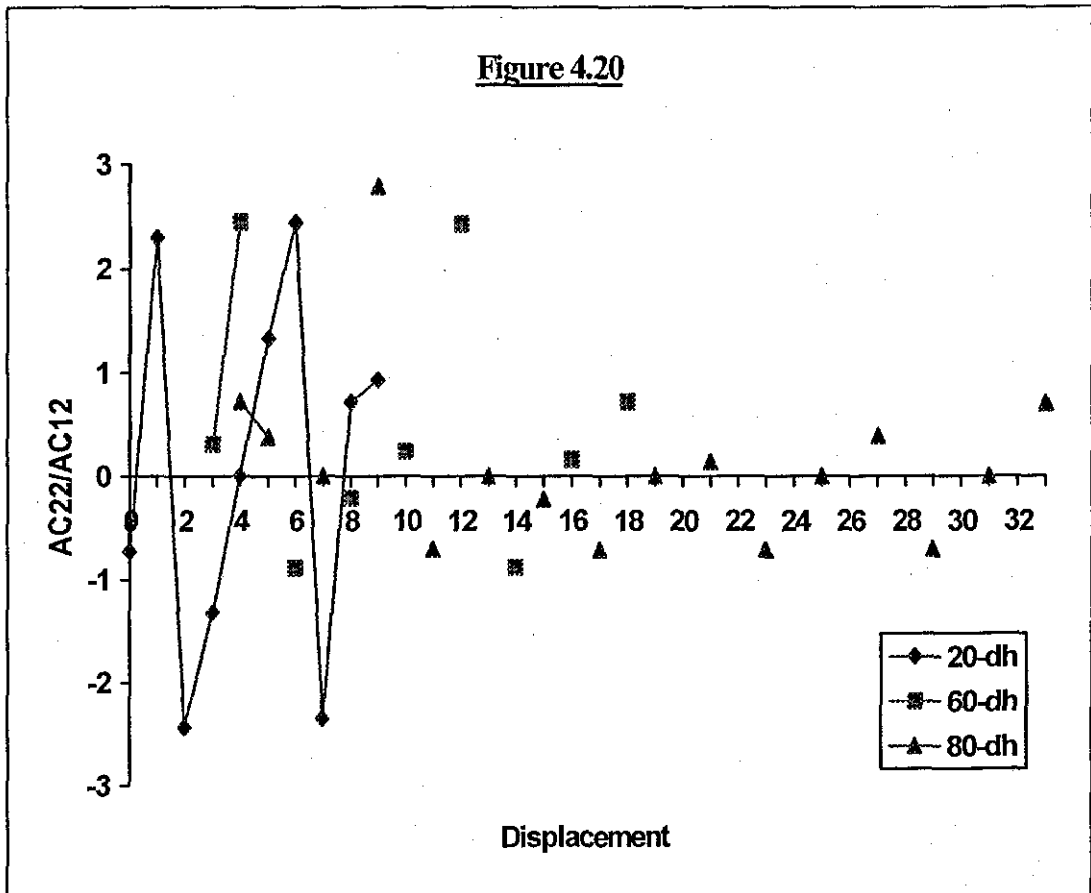


Figure 4.21

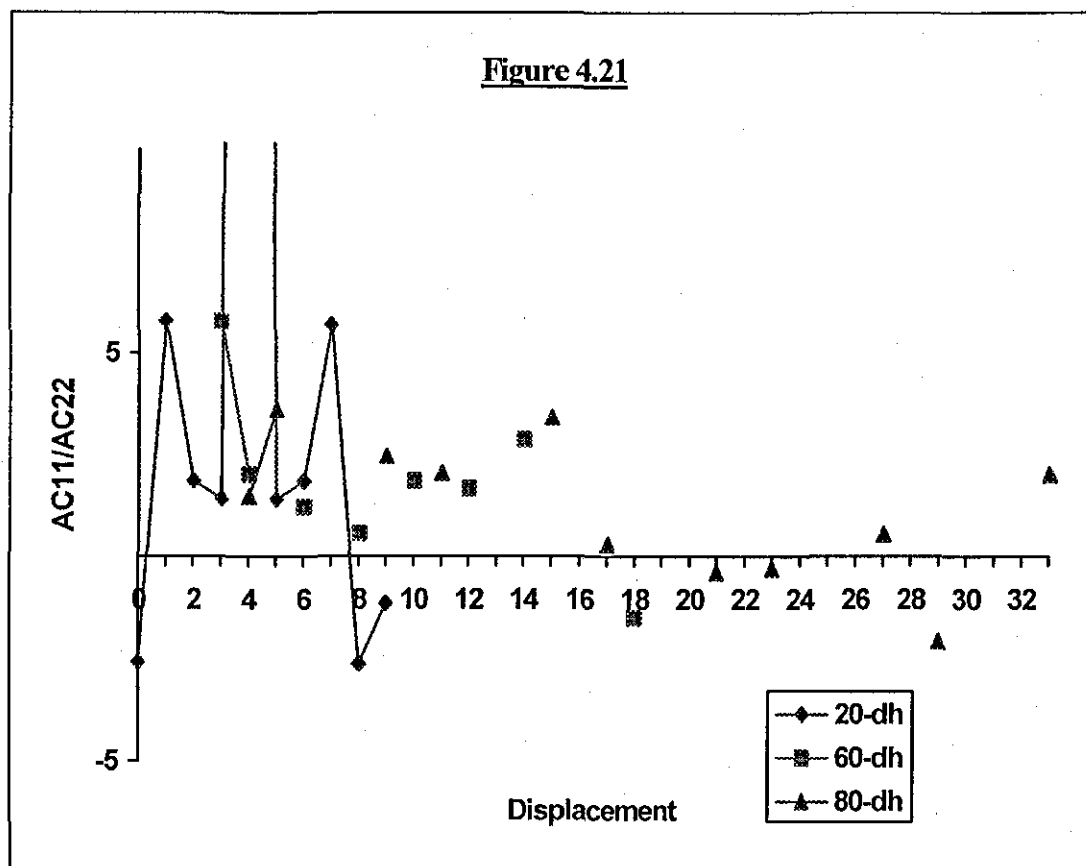
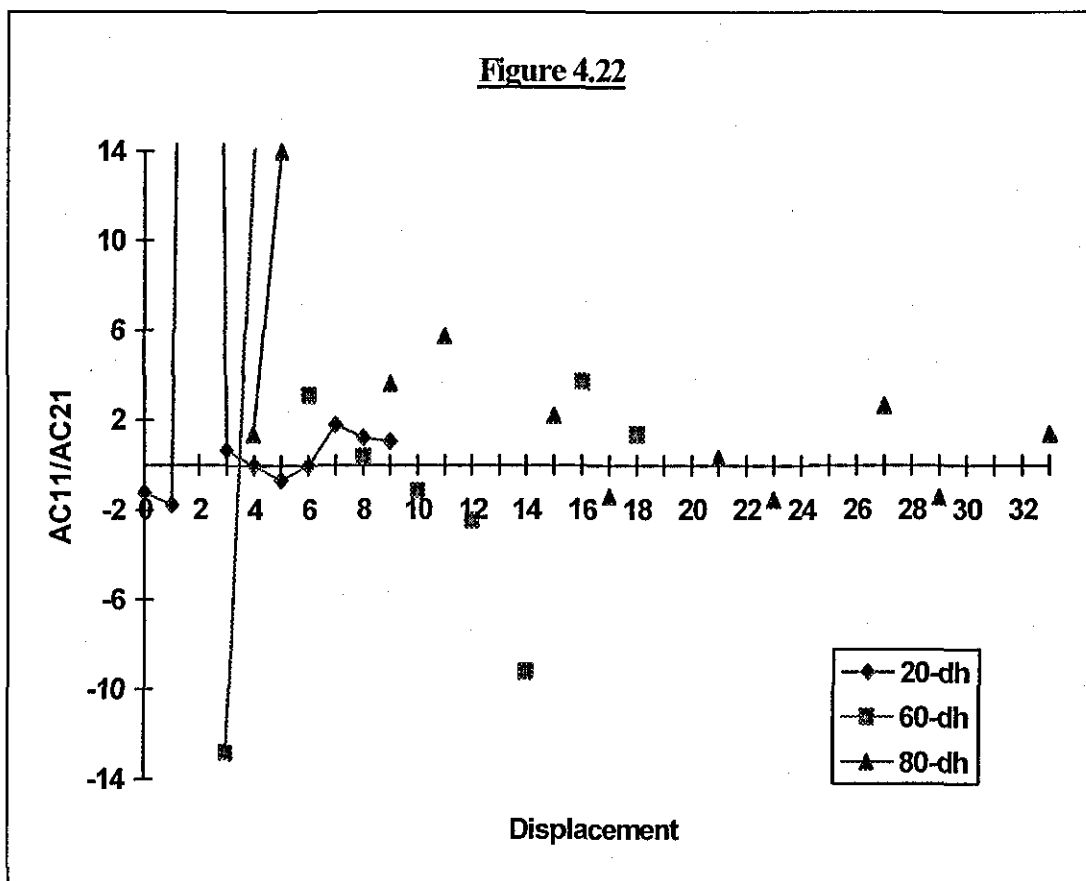


Figure 4.22



The ratios R1, R2 and R3 were chosen because for each angle and value of displacement there is a unique combination of R1, R2 and R3, its '*Transform domain ratio fingerprint*'. R2 and R3 are the main ratios used to perform the classification, however R1 gives a quick indication of which class to concentrate on, although the classification could be done with R2 and R3 alone. R1 does not contain any additional information but presents the information contained within R2 and R3 in a different manner. By using this ratio generation technique each of the surrounding blocks can be classified to have an edge at a particular angle and displacement/ position or be a flat block containing no significant features or a block containing edge information which does not fall into one of the other classes. This class is defined as unknown and blocks containing multiple edges or edges that change angle within the block are some of the possibilities, which result in such a classification. The amount of classes within the classifier has a bearing on the processing speed and accuracy of the classifier and subsequent concealment stage, however the amount of complexity remains the same i.e. the calculation of ratio values from the transform domain coefficients. Figure 4.23, 4.24 and 4.25 show the classification boundaries for classifiers with two, three and four diagonal classes (the Initial classifier having one diagonal class as seen in Figure 4.3). When an edge is classified into one of the classes within a given classifier, it is also given a value for its position within the block. This is done using the ratio information, that is for a given angle and position within a block the ratio values of a given edge will lie between certain values irrespective of the contrast between either side of the edge. This is one of the advantages of analysing ratio information rather than coefficient values.

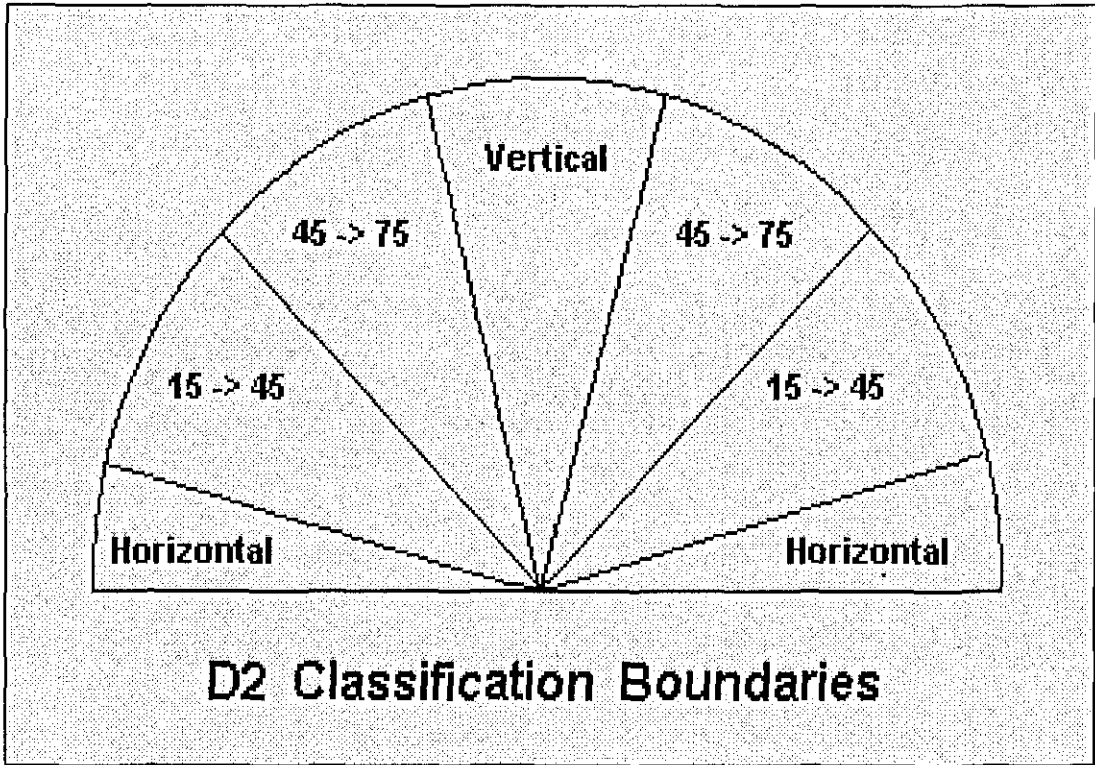


Figure 4.23

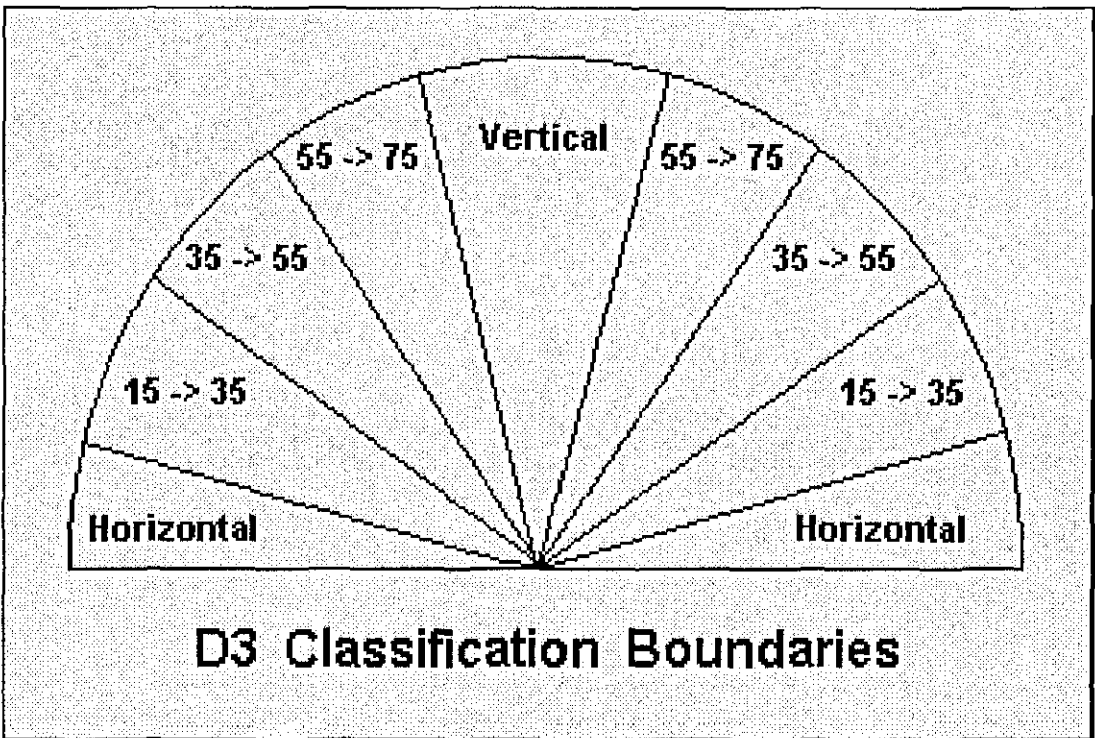


Figure 4.24

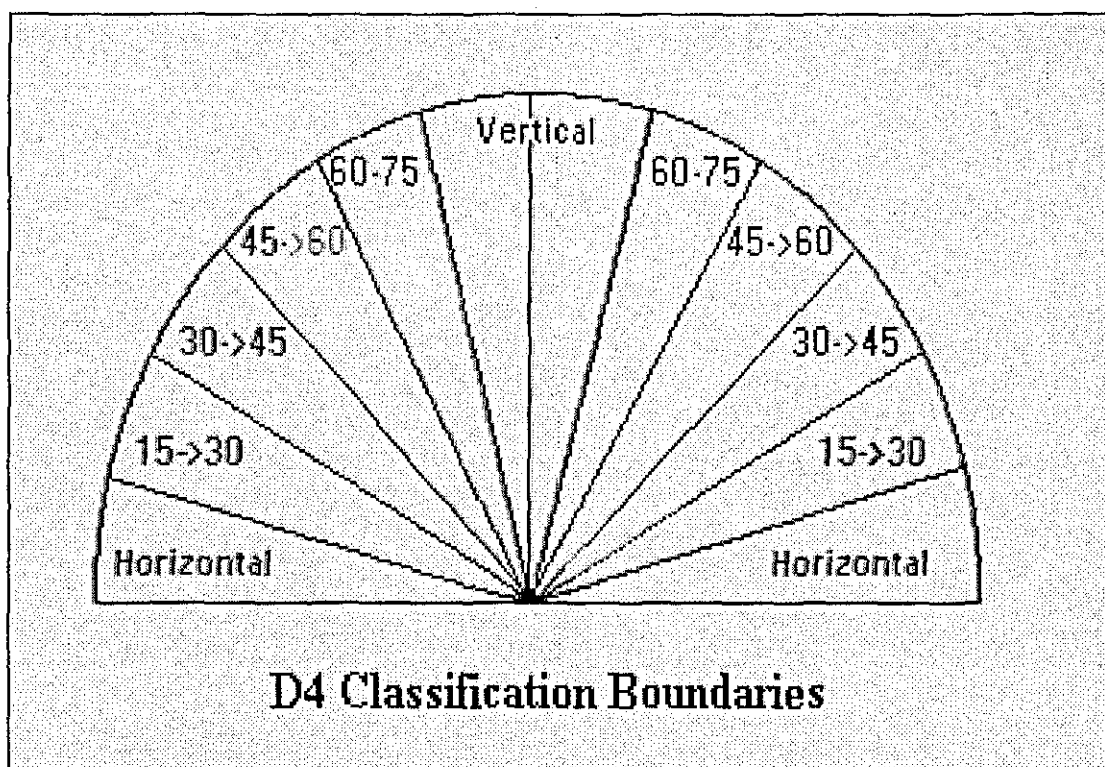


Figure 4.25

The diagonal classes were chosen by simply dividing the region between the vertical and horizontal classes. Figures 4.11, 4.12, 4.13, 4.14 and 4.15 show the characteristics for R1, R2 and R3 for edges with the angles 25, 45, 60, and 80 degrees, up hill and down hill. These results were obtained experimentally from idealised edges. To produce a usable classifier from these results, each ratio for each value of displacement and angle is given an allowable range. This technique causes the classifier to give the same result for a range of angles, and produces clearly defined classification boundaries. This process is analogous to the quantisation of a signal, with the corresponding quantisation error. As the number of classes increases the range of angles reduces within each class, as does the allowable range for the ratios and the quantisation like error. The effects of non-idealised edges are also tolerated using this technique. Figure 4.16 and 4.17 show the characteristics for R2 and R3 for a wider range of edges to show that as the angle of the edge increases the ratio graph shifts to the right and is distinct from the ratio of edges greater and smaller in angle. There are exceptions to the above, for small values of displacement the Ratio2 values are very similar due to the fact that an 8 by 8 pixel block only has a certain amount of resolution. However, in conjunction with Ratio3 the classifier is able to differentiate

between the individual edges, i.e. the edges 'Transform domain Ratio Fingerprint'.

Ratios 1 and 3 for up hill edges, bottom left to top right, are the reflection about the origin (x-axis) of those for down hill edges or have antisymmetry about the origin. In addition, Ratios R2 and R3 have antisymmetry about the origin and a central value of displacement. R1 is symmetric about the same value of displacement. Ratio 2 is the same for both up and down hill edges. The Ratios can also be represented or modelled empirically using approximations. For example, R1, R2 and R3, up hill and down hill, for 25 and 45 degree edges are shown in equations (4.1) to (4.10). In these equations n is the value of displacement and N is the maximum possible value of displacement for the particular angle ($N=9$ for 25 degree edges and $N=12$ for 45 degree edges).

$$R1(n) = 0.52 \sin \frac{\pi \cdot n}{N-1} - 0.9. \quad 25 \text{ DH.} \quad \dots\dots (4.1)$$

$$R1(n) = 0.9 - 0.52 \sin \frac{\pi \cdot n}{N-1}. \quad 25 \text{ UH.} \quad \dots\dots (4.2)$$

$$R2(n) = -1.32 \cos \frac{\pi \cdot n}{N-1}. \quad 25 \text{ DH/UH.} \quad \dots\dots (4.3)$$

$$R3(n) = 1.19 \cos \frac{\pi \cdot n}{N-1}. \quad 25 \text{ DH.} \quad \dots\dots (4.4)$$

$$R3(n) = -1.19 \cos \frac{\pi \cdot n}{N-1}. \quad 25 \text{ UH.} \quad \dots\dots (4.5)$$

$$R1(n) = -1. \quad 45 \text{ DH.} \quad \dots\dots (4.6)$$

$$R1(n) = +1. \quad 45 \text{ UH.} \quad \dots\dots (4.7)$$

$$R2(n) = -1.31 \cos \frac{\pi \cdot n}{N-1}. \quad 45 \text{ DH/UH.} \quad \dots\dots (4.8)$$

$$R3(n) = 1.31 \cos \frac{\pi \cdot n}{N-1}. \quad 45 \text{ DH.} \quad \dots\dots (4.9)$$

$$R3(n) = -1.31 \cos \frac{\pi \cdot n}{N-1} \quad 45 \text{ UH.} \quad \dots\dots (4.10)$$

Figures 4.26, 4.27, 4.28, 4.29 and 4.30 show graphically how the empirically modelled equations compare with the Ratio1, 2 and 3 experimental values for 25 and 45 degree edges. Figure 4.28 shows Ratio3 for a 25 degree edge, the modelled equation is not as good an approximation as for Ratio1 and 2 for 25, 45 and Ratio3 for a 45 degree edge. Therefore, a lookup table type of approach could be used for this Ratio value. Alternatively an algorithmic approach could be used where an offset is subtracted for values of displacement less than 4, but not 0, and added for values of displacement greater than 4 but not 8, for down hill. The opposite arrangement would be used for the up hill Ratio values. The results are also show in Figure 4.28.

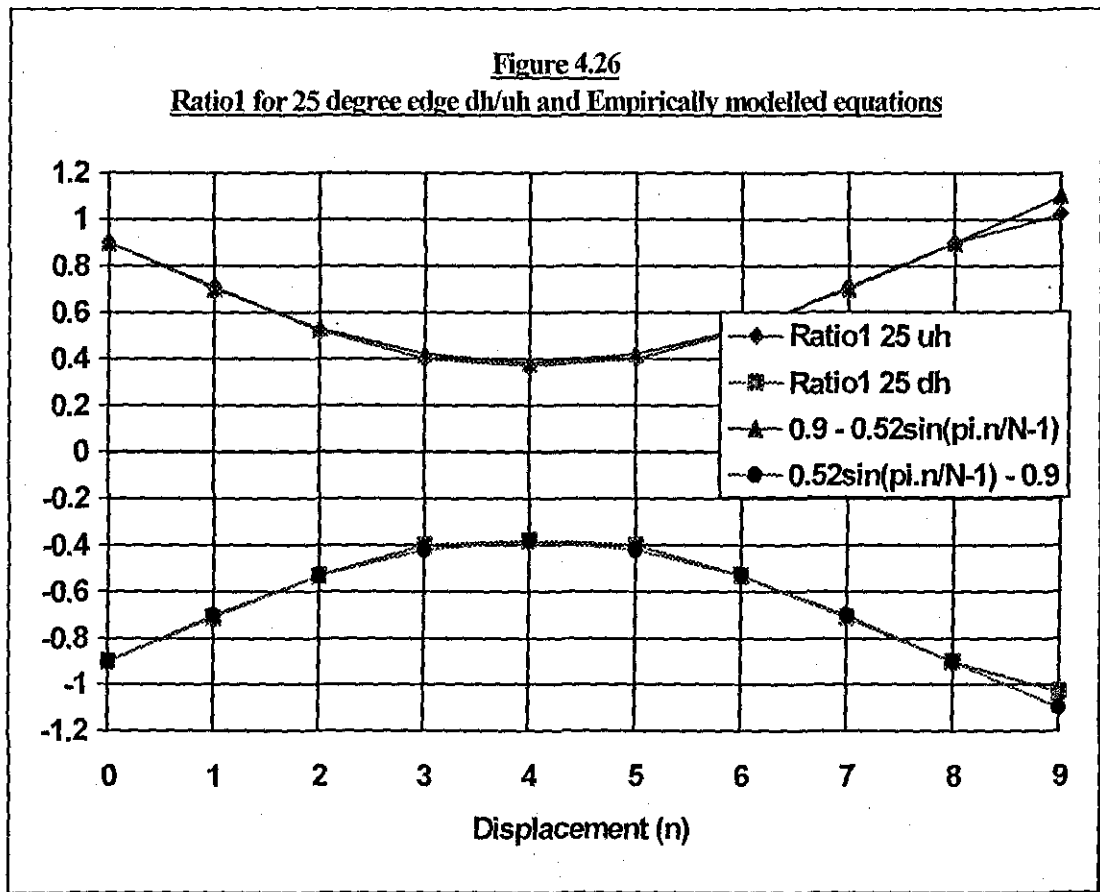


Figure 4.27
Ratio2 for 25 dh/uh and Empirically modelled equation

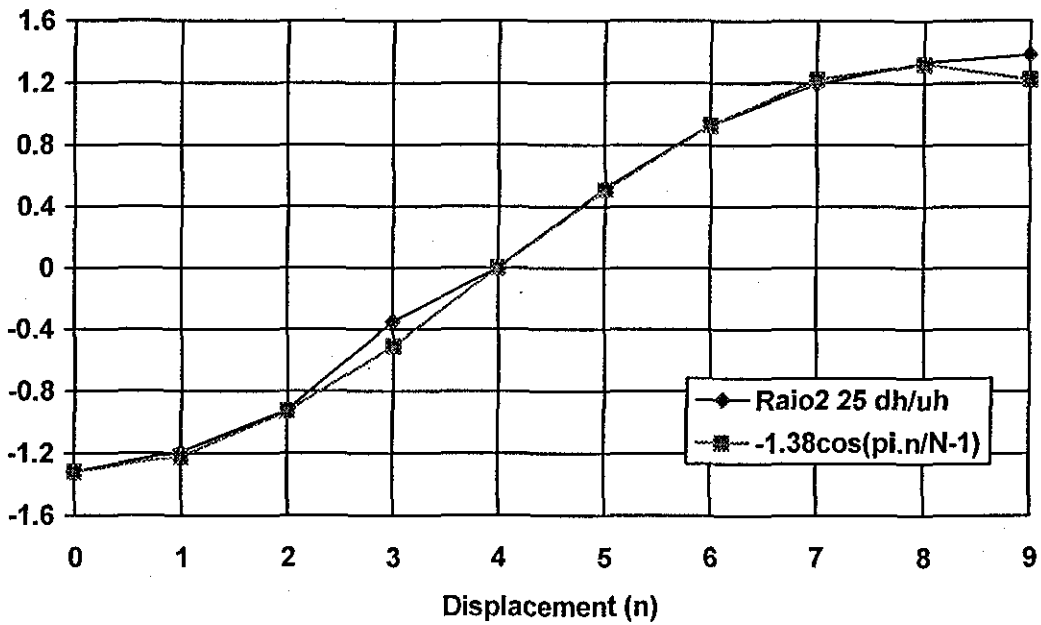


Figure 4.28
Ratio3 for 25 dh/uh and Empirically modelled equations/algorithm

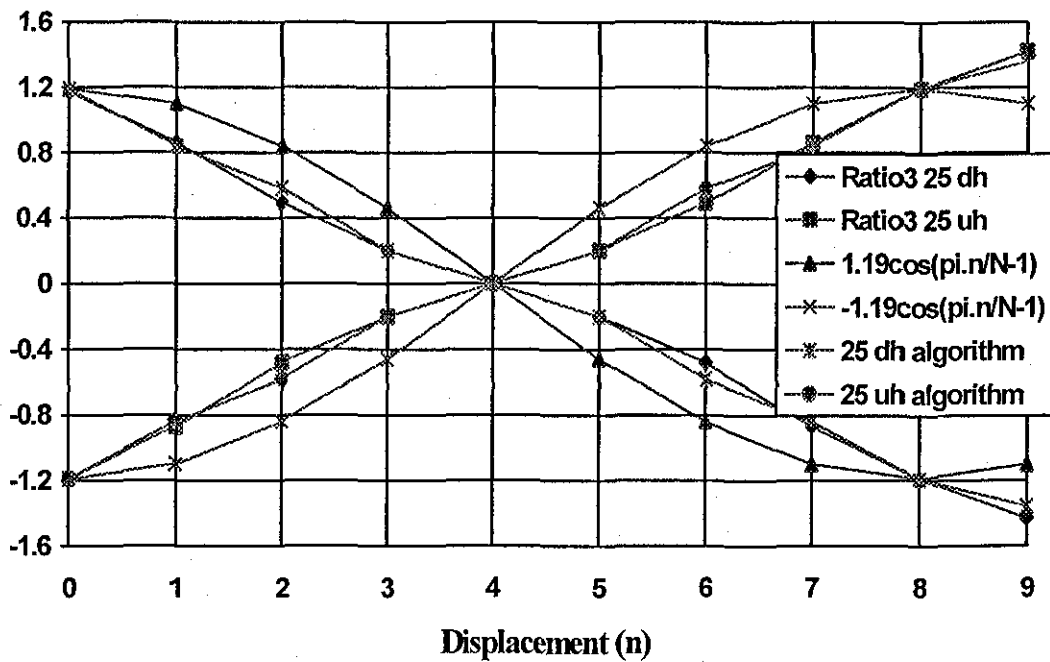


Figure 4.29
Ratio2 for 45 dh/uh and Empirically modelled equation

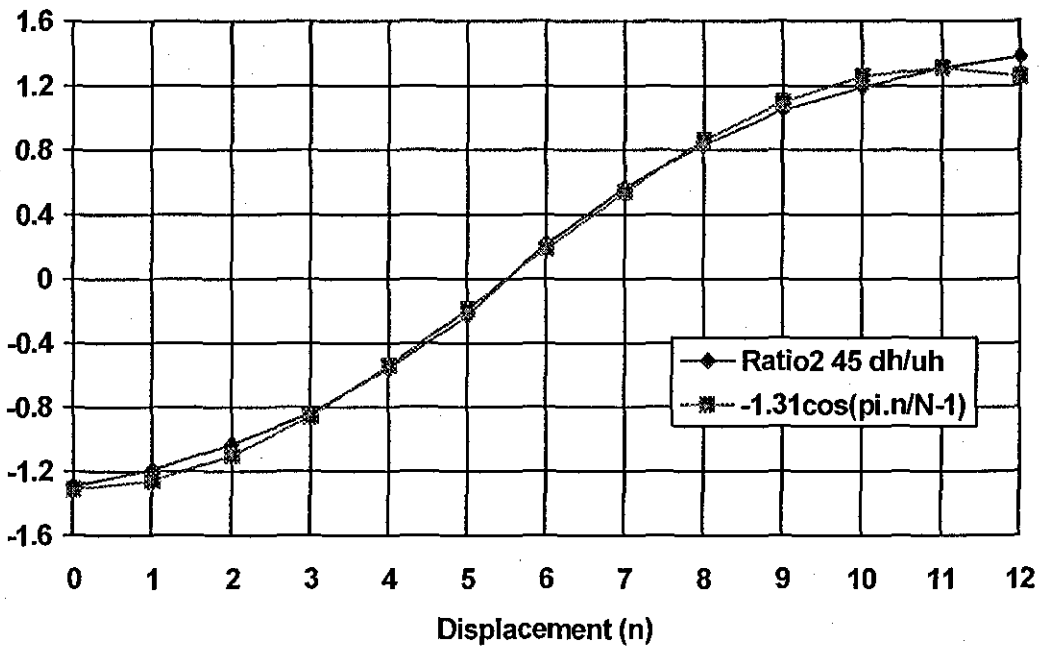
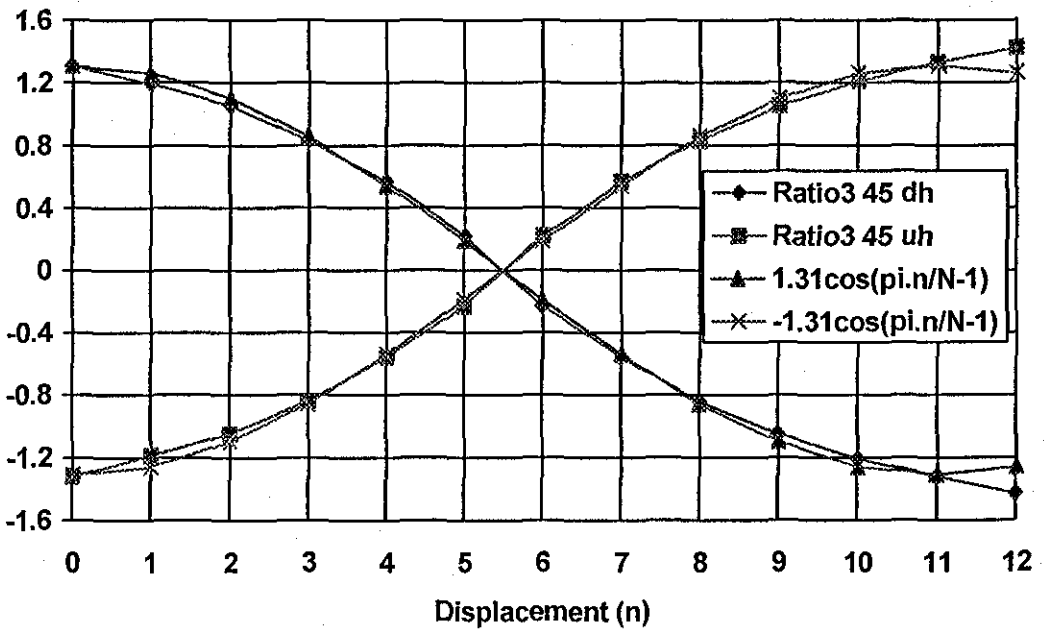


Figure 4.30
Ratio3 for 45 dh/uh and Empirically modelled equations



In the classification process, the Ratio values are calculated from the transform domain coefficients and the angle and displacement/position of the edge is required. Therefore the classifiers for D2, D3 and D4 use an algorithmic approach to find the correct angle and displacement classification. The Ratio values obtained for the surrounding block are compared with the values of the classification boundaries. If R2 and R3 fall within the boundaries for a particular angle and value of displacement, then these two values are the result given by the classification algorithm. If R2 is within the boundaries but R3 falls outside the boundaries then this particular value of displacement is incorrect, and the next value of displacement is then tested. After all the possible values of displacement for a particular class has been tested, the classifier begins to check the ratio values against those for the next class within the specified classifier. Once all the classes have been checked, the algorithm will class the particular block as an unknown block if no match has been found. The classifier first checks to see if the block is a Flat block containing no significant features. The empirically modelled equations can be used in the concealment stage to calculate the required Ratio values for an edge at a particular angle and displacement required to conceal the error block. Once the Ratio values have been found, the low frequency coefficient values can be calculated.

4.4 Classification Results.

The following graphs, Figures 4.31, 4.32, 4.33 and 4.34 show the classification results for the classifiers D1, D2, D3 and D4, for a sequence of 13 frames with 9% errors. Since some of the edges were on the edge of the image not all errors will have eight surrounding blocks. The amount of Flat classifications is not shown in the graphs to show the detail of the edge classifications. Also the classifiers for D2, D3 and D4 used R2 and R3 only to produce the classifications, the D1 classifier used the Initial Classification Algorithm [3]. Each classifier D1, D2, D3 and D4 produced the same amount of Flat classifications i.e. 2514 blocks. The graphs show that the D1 classifier has classed some blocks as diagonal that D2, D3 and D4 have classed as unknown. This shows that classifying blocks by coefficient magnitudes produces a number of incorrect classifications.

Figure 4.31
D1 Initial classifier

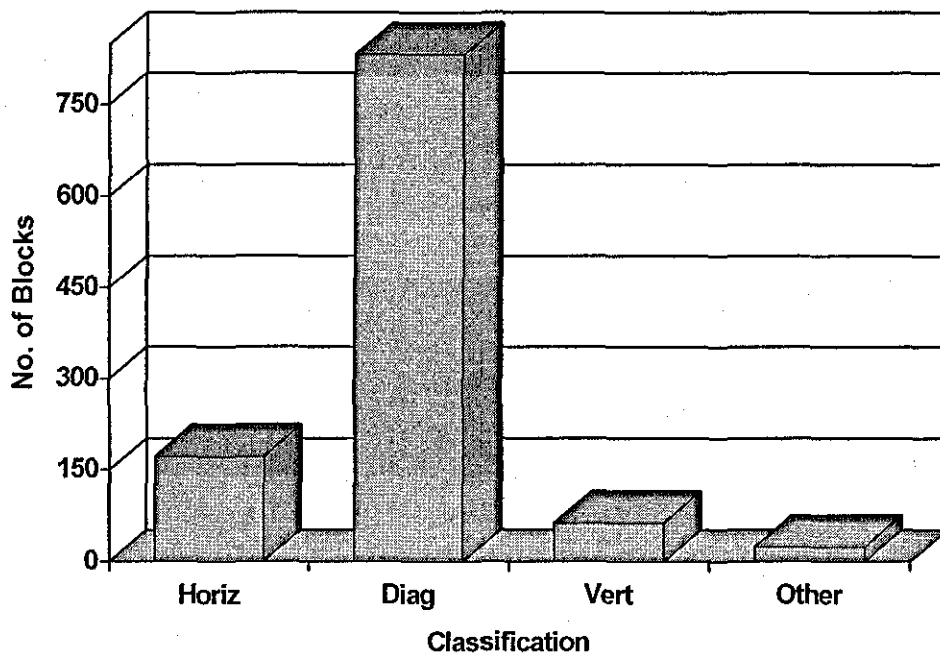


Figure 4.32
D2 classifier

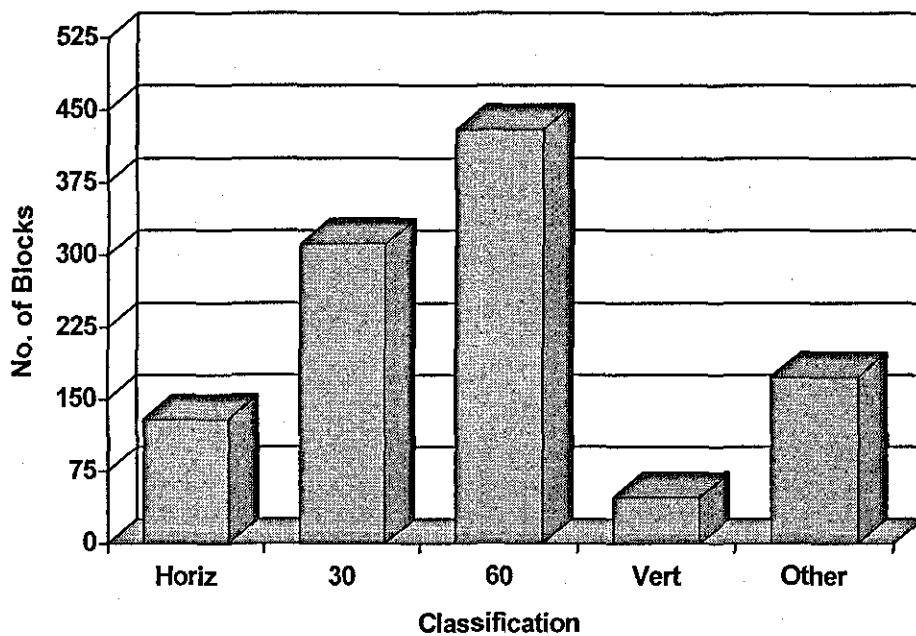


Figure 4.33
D3 classifier

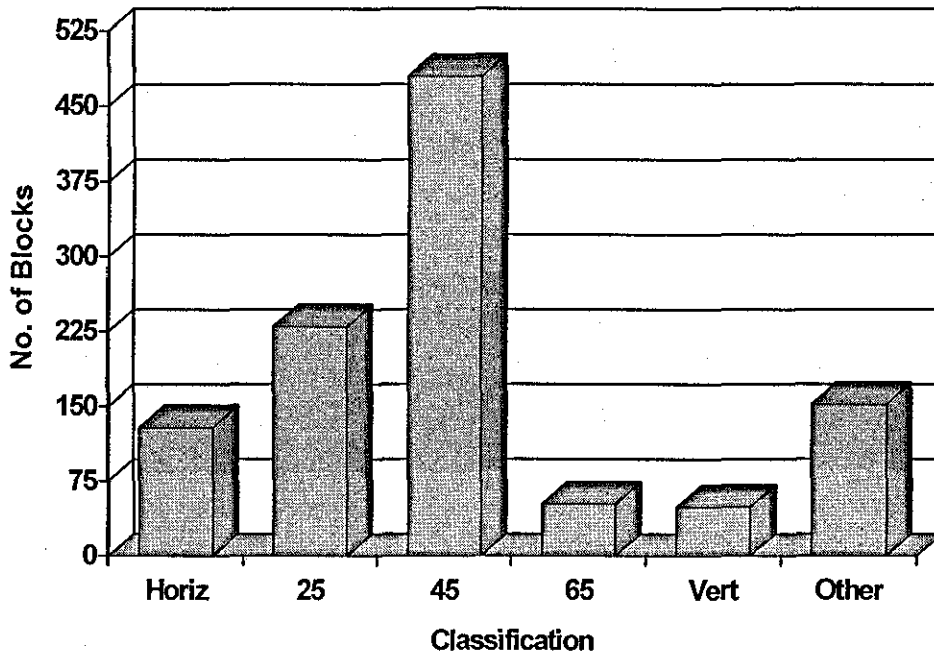
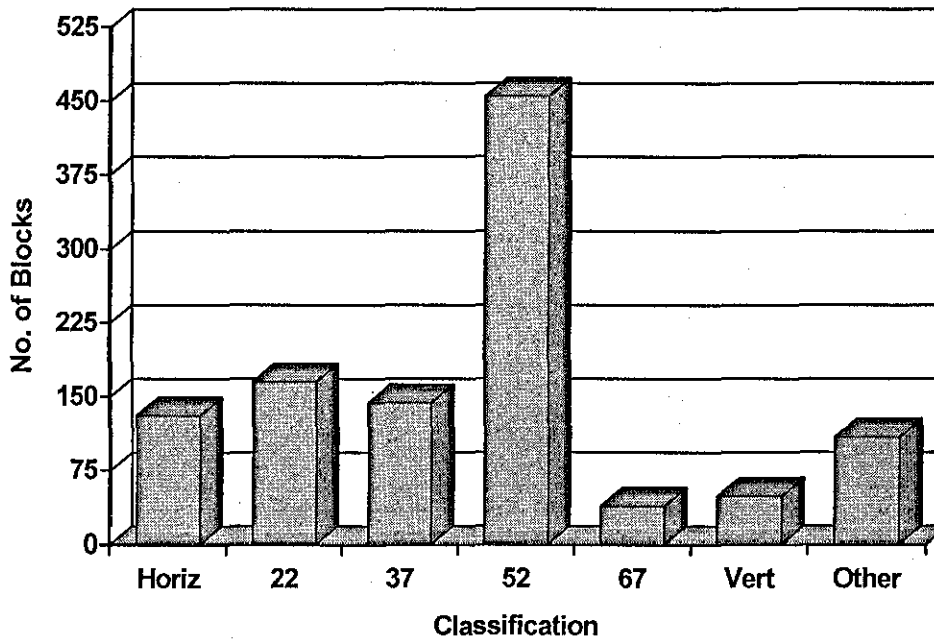


Figure 4.34
D4 classifier



4.5 Summary and Conclusions.

This Chapter has introduced the idea and algorithms to perform Feature Detection and Edge Classification within the transform domain. An investigation into the representation of edges in the transform domain was performed to identify the individual '*fingerprints*' of a range of Features/Edges. From an Initial Classifier where the classification was performed using the magnitude of certain coefficients, an Advanced Classifier was developed. The classification of edges using the Ratio generation technique overcomes the failings of the Initial Classification method, also additional information is obtained i.e. an estimate of the edges position and angle within the block. Consequently, the concealment of missing blocks is enhanced as a result of the availability of the position and angle information.

An advantage of this form of classification is the minimal amount of processing required, 2 or 3 divisions and a number of comparisons, irrespective of block size and amount of classes within a given classifier. That is the classification processing load is not directly proportional to the block dimensions as can be the case with pixel domain methods [6, 22, 23, 25, 26, 27, 28].

CHAPTER 5

CHAPTER 5:

TRANSFORM DOMAIN ERROR CONCEALMENT.

Error Concealment is the process of deriving approximations for lost data from the information in the received data stream. There are a number of alternative approaches or techniques to concealment, Lam and Reibman [37] proposed an algorithm to conceal bit errors where the detection was performed in the transform domain and concealment in the pixel domain. Also Wang and Zhu [38] proposed an algorithm to conceal for the loss of individual coefficients. However, within this research a bit error or the loss of a coefficient would cause an MCU/ MB to be dropped or discarded causing the loss of a number of DCT blocks.

Previous work at Loughborough has shown the feasibility and advantages of error concealment in the transform domain before image reconstruction as opposed to the conventional pixel domain processing after reconstruction [6, 22, 23, 25, 26, 27, 28]. Considerable improvements in processing speed and simplicity in block classification have resulted [29, 30]. The amount of processing required to perform concealment can be greatly reduced when concealing in the transform domain due to the fact that an 8 x

8 pixel block can be accurately represented by a small number of DCT coefficients. As shown in Chapter 4, only a small number of coefficients, located in the top left-hand corner of the transform domain block, are significant. This is due to the '*energy compaction property*' of the Discrete Cosine Transform, a property that is taken advantage of in the encoding/compression process. The calculation of say 8 coefficients can be carried out significantly faster than the calculation of 64 pixels. This research expands on the simple Feature Detection/ Concealment ideas and classification of blocks within the transform domain, introduced in [31], to aid the development of an advanced Transform Domain Feature Detection, Classification and Concealment algorithm, that requires the calculation of only 4 transform domain coefficients.

The concealment of a lost block when using Linear Interpolation of the transform domain coefficients produces satisfactory results when the error block contains little or no edge information. However, the results when an edge is present within the error block and its surrounding blocks are very poor and visual discontinuities are present within the concealed edge. These discontinuities are instantly apparent to a viewer, especially if the image is one of a sequence of images/frames as in MJPEG or MPEG. Therefore, an improved method of concealment is required which makes use of the edge information in the surrounding blocks, to aid concealment.

Once the surrounding blocks have been classified the concealment algorithm inspects the results and if an edge is found with displacement such that the edge will enter the error block, the algorithm deduces the displacement required for the edge in the error block. Once the value of displacement is found the algorithm then calculates the four coefficients $DC00(n)$, $AC01(n)$, $AC10(n)$ and $AC11(n)$ required at that position, for the Luminance block, so that the edge is continued through the error block. The algorithm effectively shifts the edge in the surrounding block to the calculated value of displacement, such that when placed in the error block it produces a continuous edge. Only three AC coefficients are calculated because for blocks containing single edges the majority of the AC energy required to represent the edge in the transform domain is concentrated in the three chosen coefficients. Therefore, the improvements gained by calculating additional AC coefficients are minimal and do not justify the

extra processing required. Also, due to the quantisation process higher frequency coefficients are often absent. The remaining AC coefficients are set to zero. The 'Mach Band' effect [36] compensates somewhat for the lack of high frequency coefficients. Figures 5.1 to 5.15 show how the DC00(n), AC01(n), AC10(n) and AC11(n) coefficients of the Luminance block vary as 25, 45, 60 and 80 degree, down hill edges, are displaced across a block. Most Figures have a number of example graphs representing edges with different values of contrast on either side of the edge.

As can be seen from Figures 5.1 to 5.16, the shape of the graphs are approximately the same for all five examples of contrast. The characteristics for coefficients DC00(n), AC01(n), AC10(n) and AC11(n) for all other edges have similar results. Also, the coefficients for 80 degree edges are more complex than those for 25, 45 and 60 degree edges, especially coefficients AC10(n) and AC11(n), as can be seen in Figures 5.8, 5.12, 5.16. This observation was another contributing factor for choosing 75 degrees as the upper boundary for the class nearest the Vertical in each classifier D2, D3 and D4.

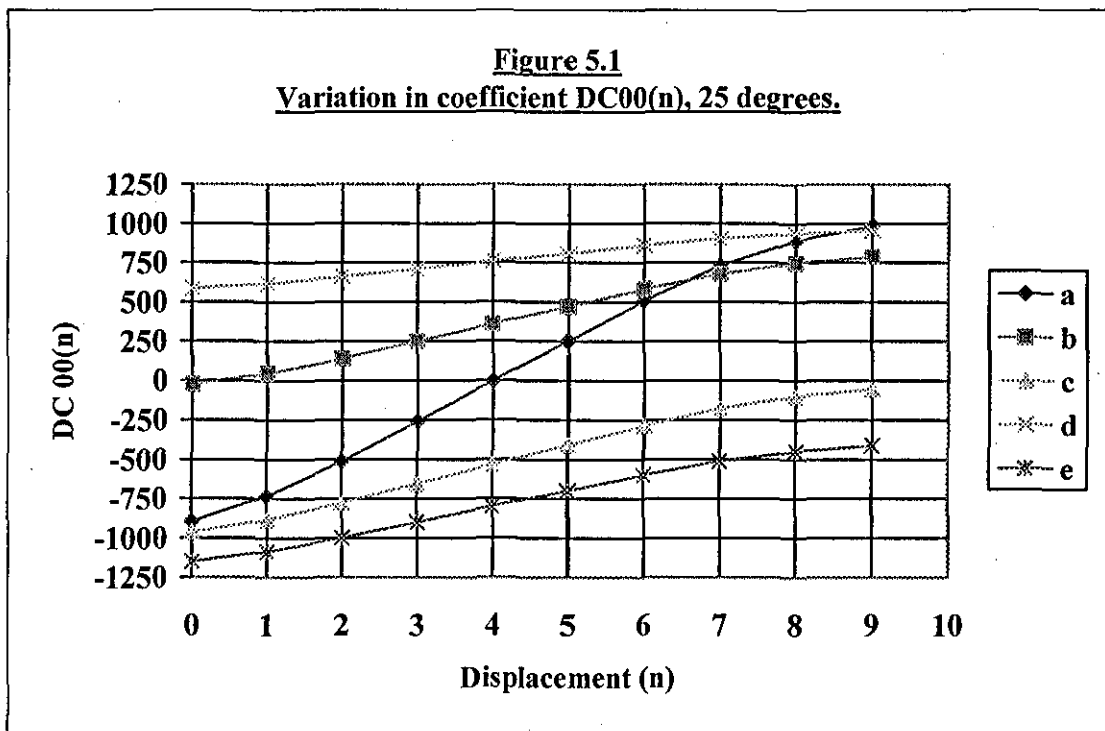


Figure 5.2
Variation in coefficient DC00(n), 45 degrees.

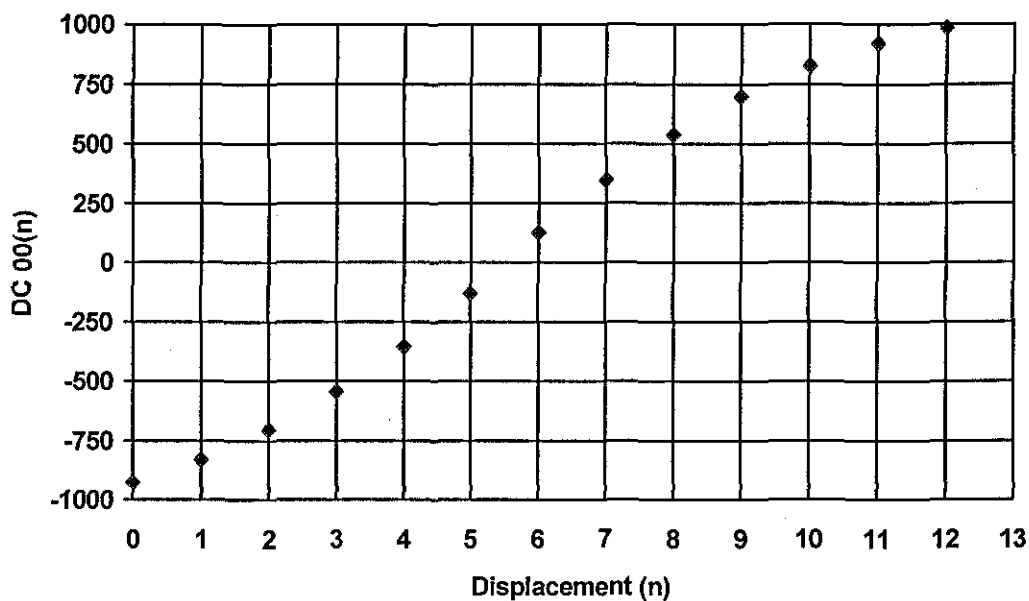


Figure 5.3
Variation in coefficient DC00(n), 60 degrees.

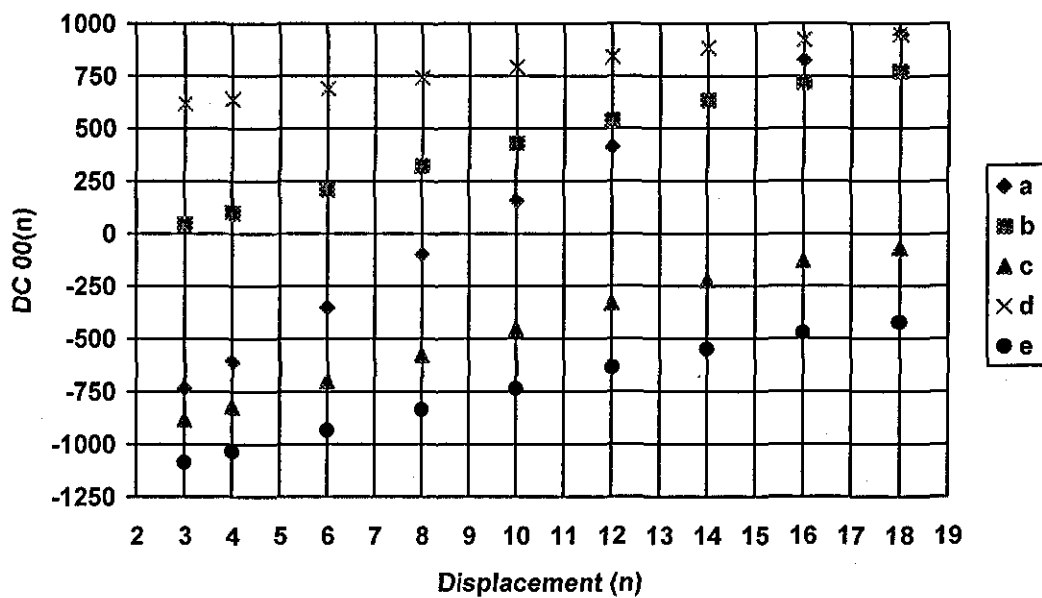


Figure 5.4
Variation in coefficient DC00(n), 80 degrees.

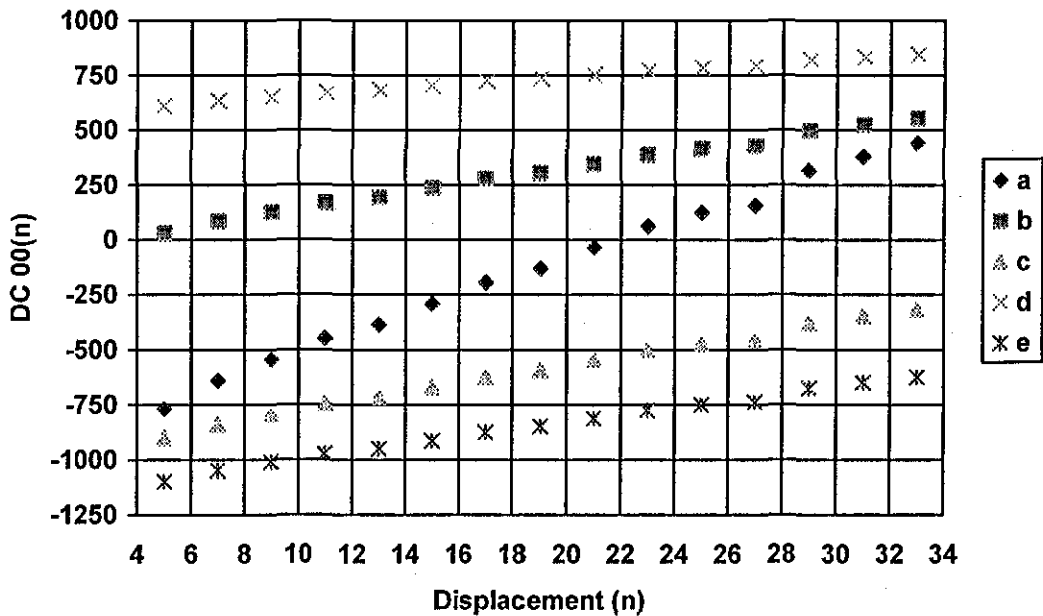


Figure 5.5
Variation in coefficient AC01(n), 25 degrees.

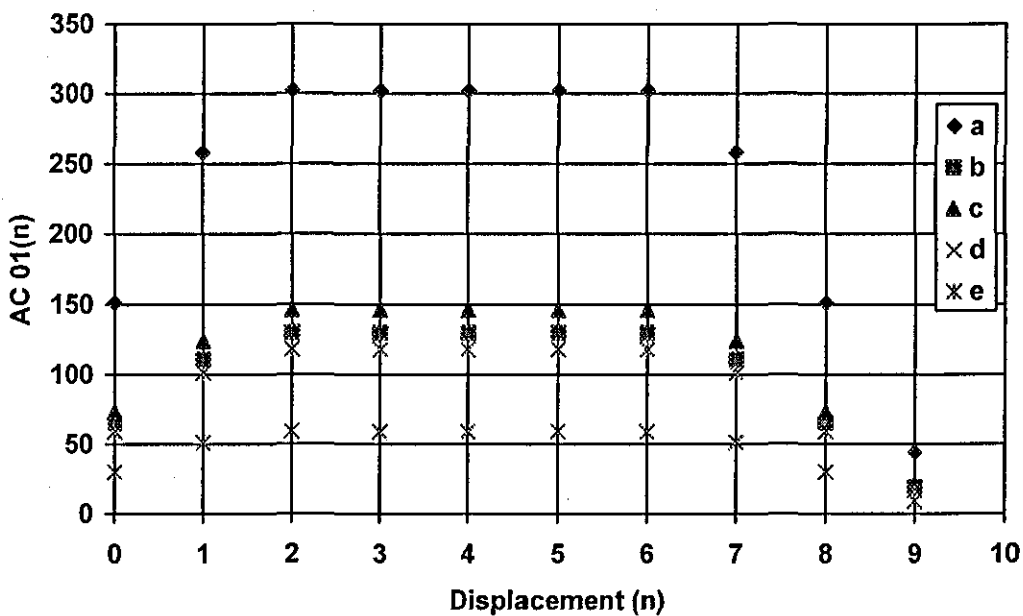


Figure 5.6
Variation in coefficient AC01(n), 45 degrees.

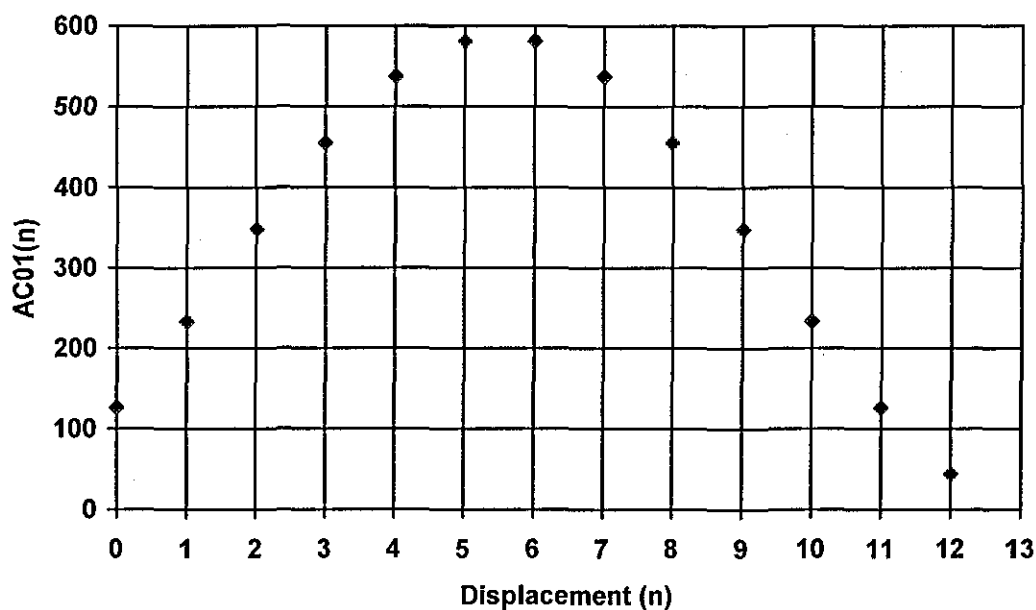


Figure 5.7
Variation in coefficient AC01, 60 degrees.

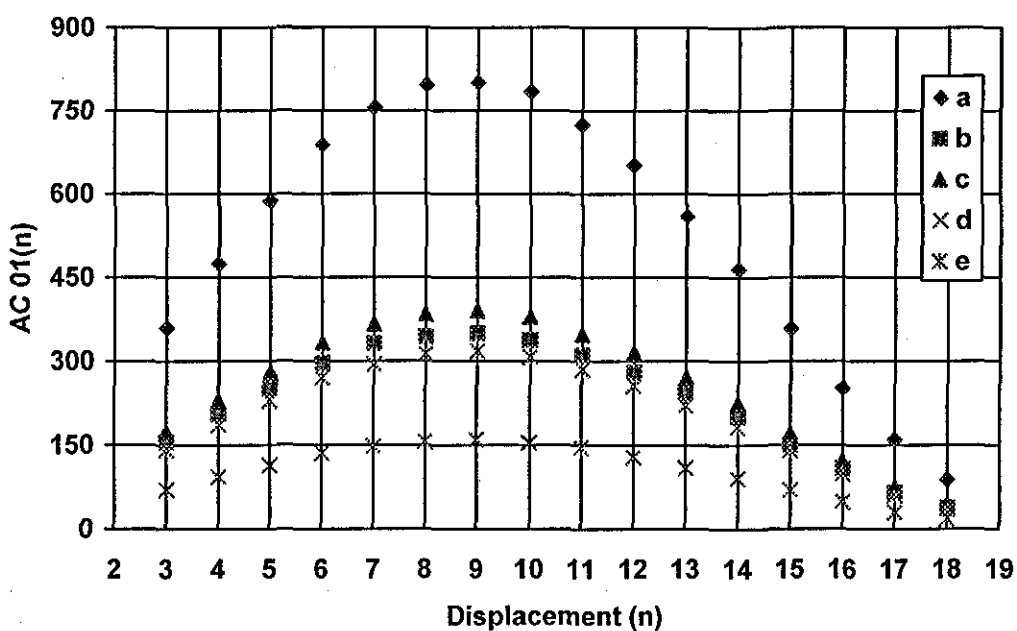


Figure 5.8
Variation in coefficient AC01(n), 80 degrees.

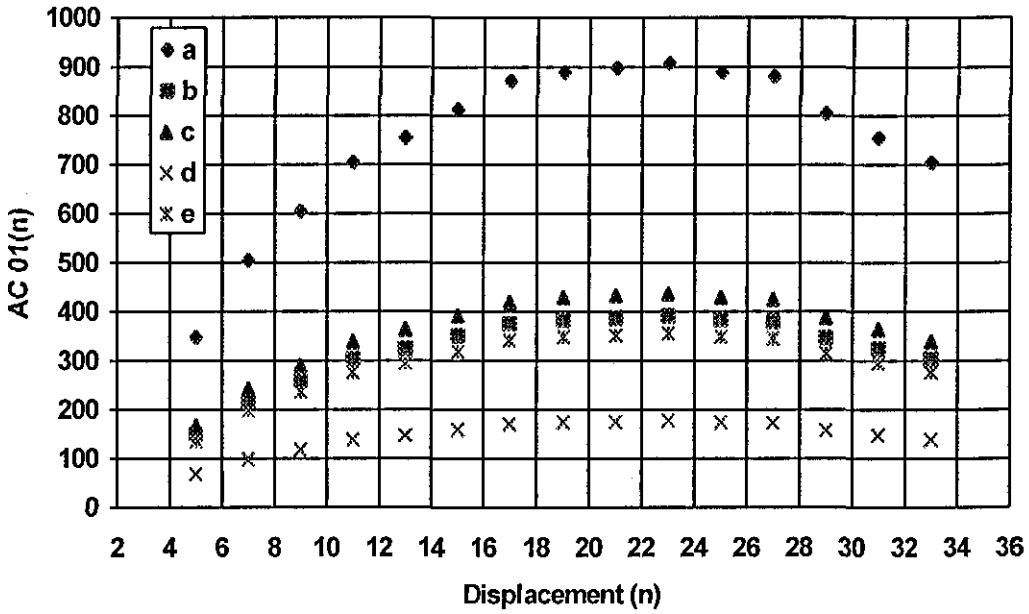


Figure 5.9
Variation in coefficient AC10(n), 25 degrees.

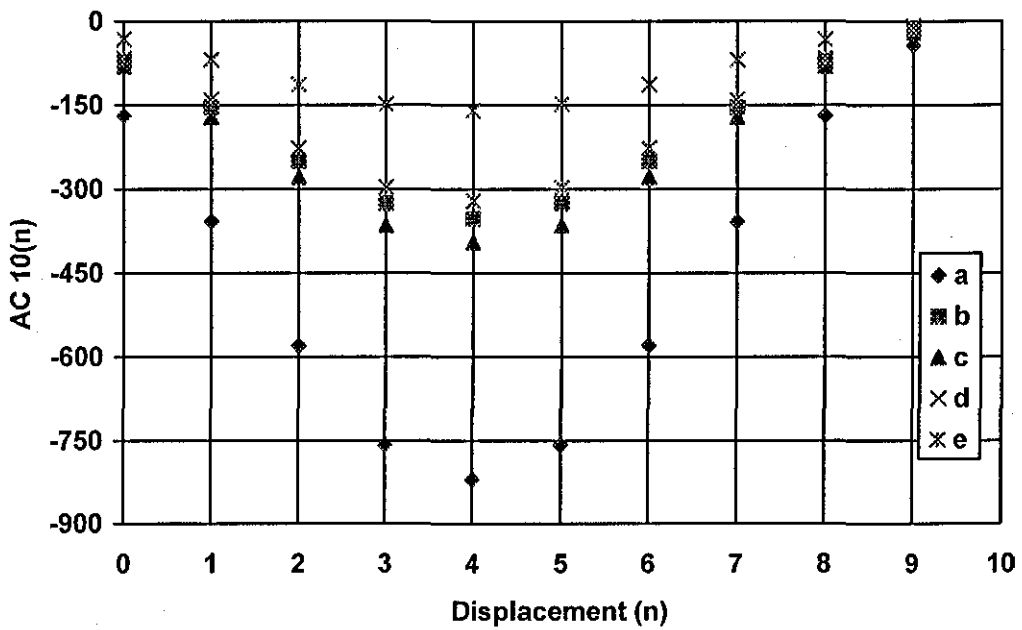


Figure 5.10
Variation in coefficient AC10(n), 45 degrees.

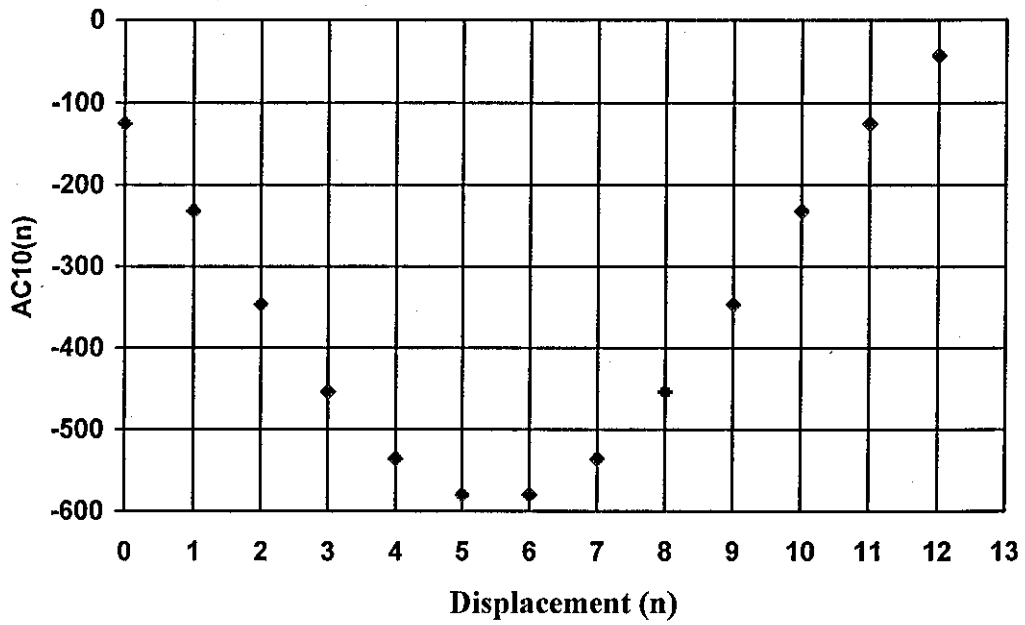


Figure 5.11
Variation in coefficient AC10(n), 60 degrees.

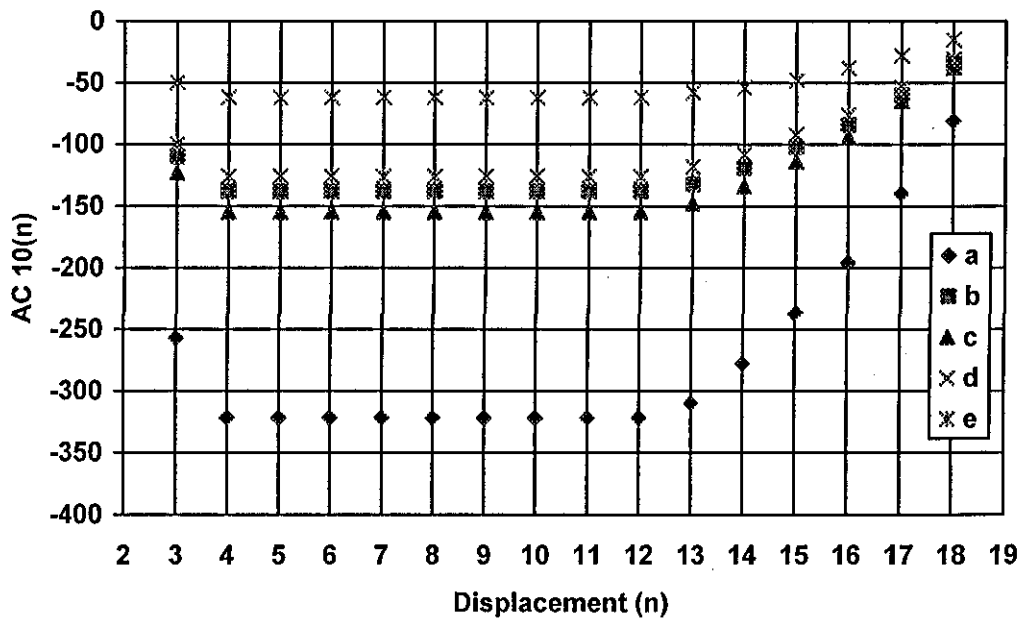


Figure 5.12
Variation in coefficient AC10(n), 80 degrees.

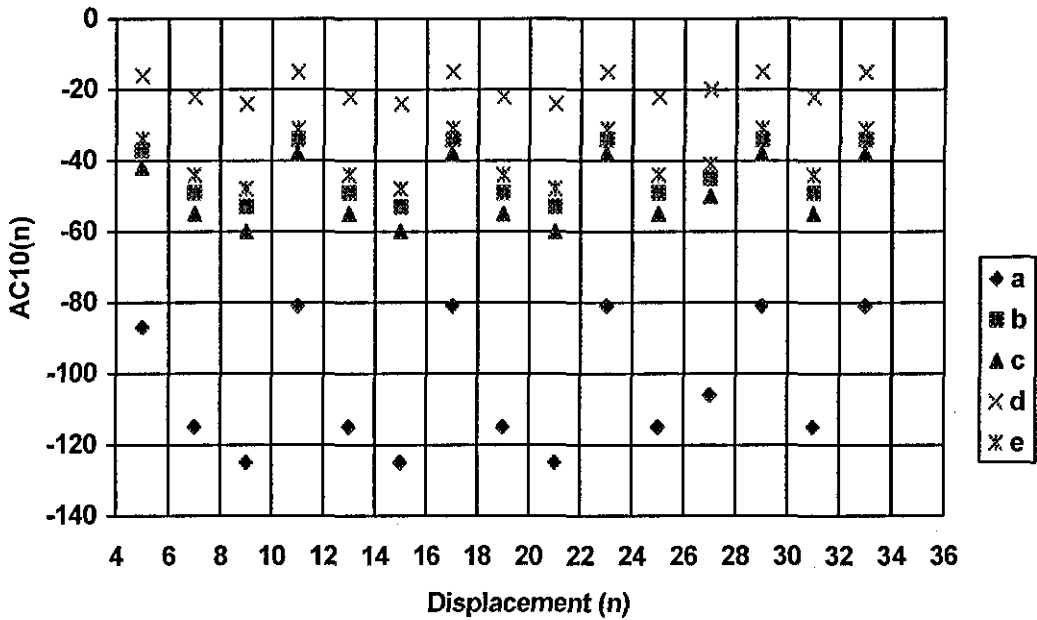


Figure 5.13
Variation in coefficient AC11(n), 25 degrees.

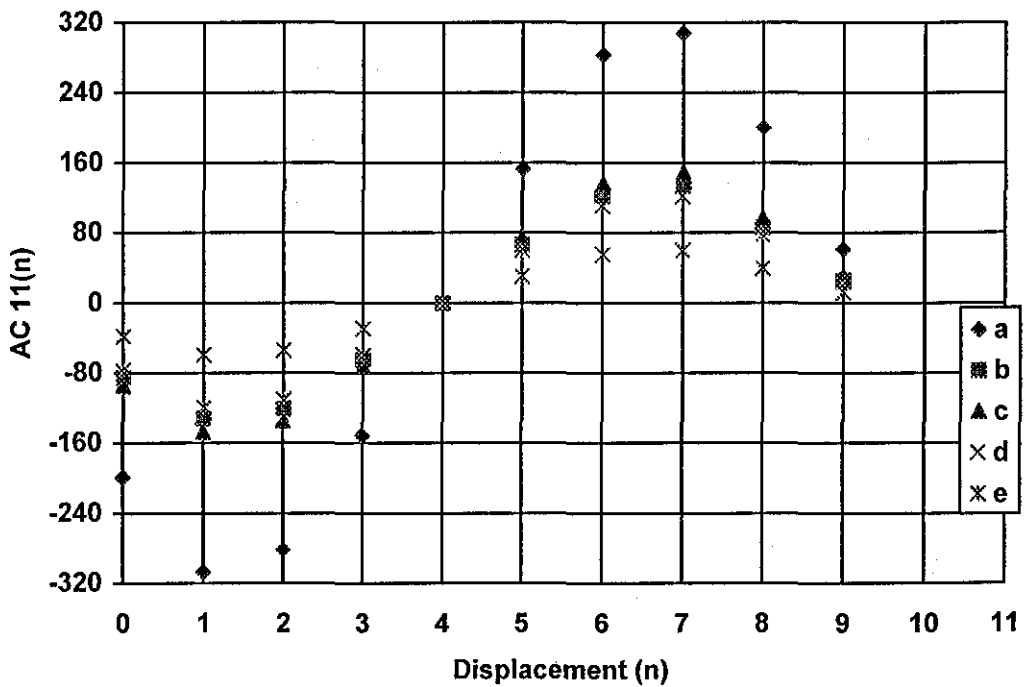


Figure 5.14
Variation in coefficient AC11(n), 45 degrees.

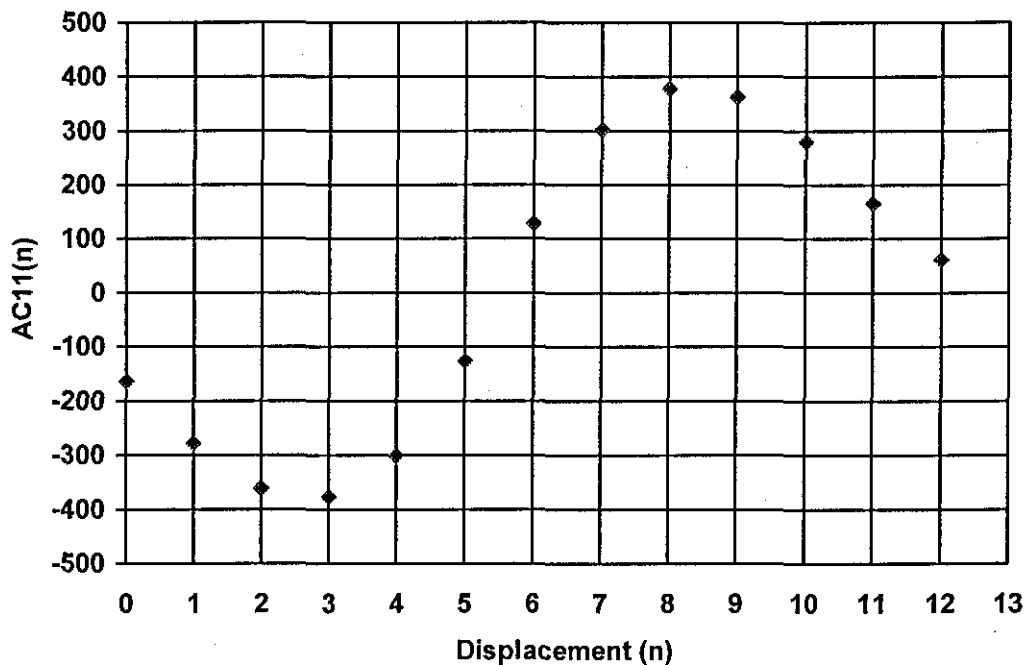


Figure 5.15
Variation in coefficient AC11(n), 60 degrees.

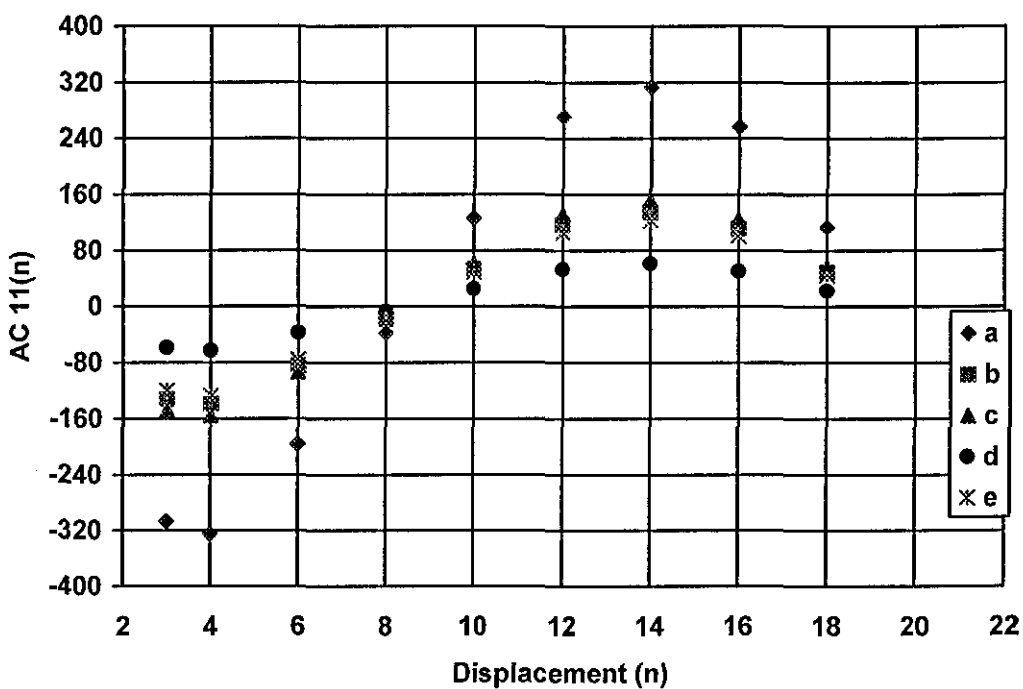
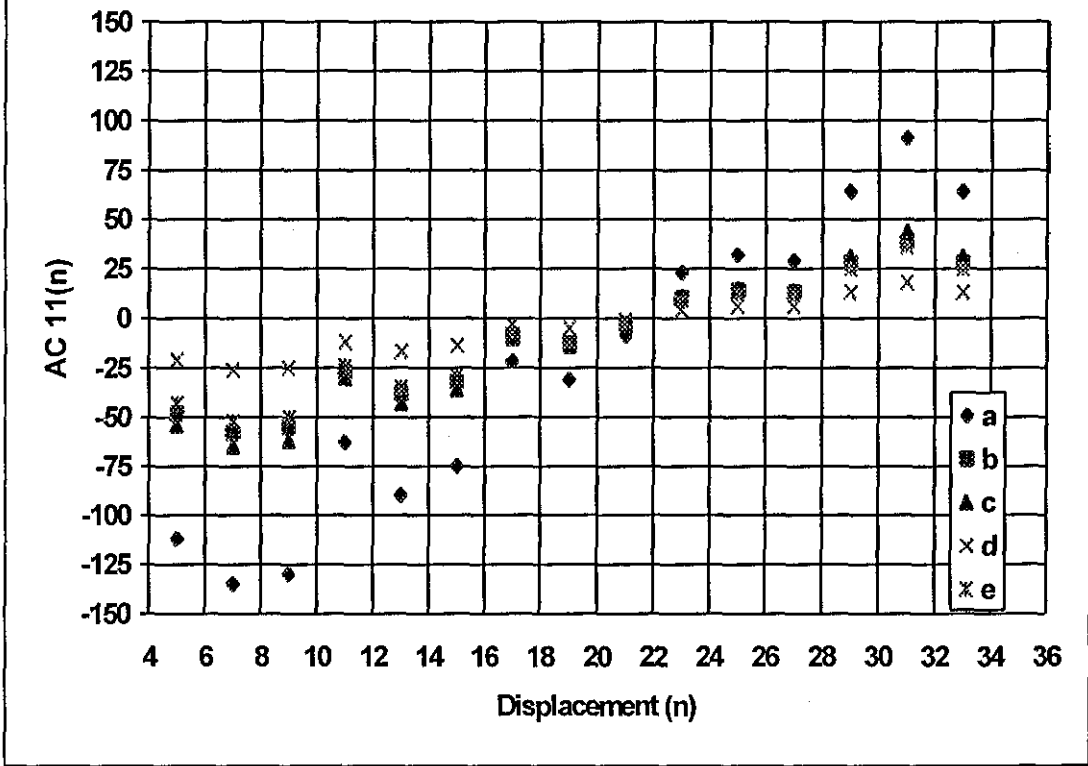


Figure 5.16
Variation in coefficient AC11(n), 80 degrees.



Most of the coefficients can be represented or mathematically modelled by simple sinusoidal or straight-line functions, as shown in equation (5.1) to (5.9), for 25, 45, 60 and 80-degree edges. The graphs show the coefficients for down hill edges, the characteristics for up hill edges are the same but are negated in most cases. In the following empirically modelled equations, N represents the maximum displacement, (n) the value of displacement, X the Peak value, Y a constant, M the gradient and C the value the graph crosses the y axis.

25 degree edge

$$DC00(n) = Mn + C. \dots\dots\dots (5.0)$$

$$AC10(n) = X \cos \frac{(2\pi(n+2))}{N-1} - Y. \quad \dots (5.1)$$

$$AC11(n) = -X \sin \frac{(2\pi(n+2))}{N-1}. \quad \dots (5.2)$$

45 degree edge

$$AC01(n) = X \sin \frac{(\pi(n+0.5))}{N-1} + Y. \quad \dots (5.3)$$

$$AC10(n) = -X \cos \frac{(\pi(n+0.5))}{N-1} - Y. \quad \dots (5.4)$$

$$AC11(n) = -X \sin \frac{(2\pi(n+0.5))}{N-1}. \quad \dots (5.5)$$

60 degree edge

$$AC01(n) = X \sin \frac{(\pi(n+2.5))}{N-1} + Y. \quad \dots (5.6)$$

$$AC11(n) = -X \sin \frac{(2\pi(n+2))}{N-1}. \quad \dots (5.7)$$

80 degree edge

$$AC01(n) = X \sin\left(\frac{\pi \cdot n}{N-1}\right) + Y \dots\dots\dots (5.8)$$

$$AC11(n) = -X \sin\left(\frac{2\pi(n+8)}{N-1}\right) \dots\dots (5.9)$$

The equations for DC00(n) for 45, 60 and 80 degrees are the same as for 25 degrees with different values for M and C. Figure 5.17 shows that a straight line function models DC00(n) better than a cosine function, for an edge with a difference in contrast of 255 between either side of the edge i.e. pixel values of 127 and -128. This is the case for all contrast differences, for all angles.

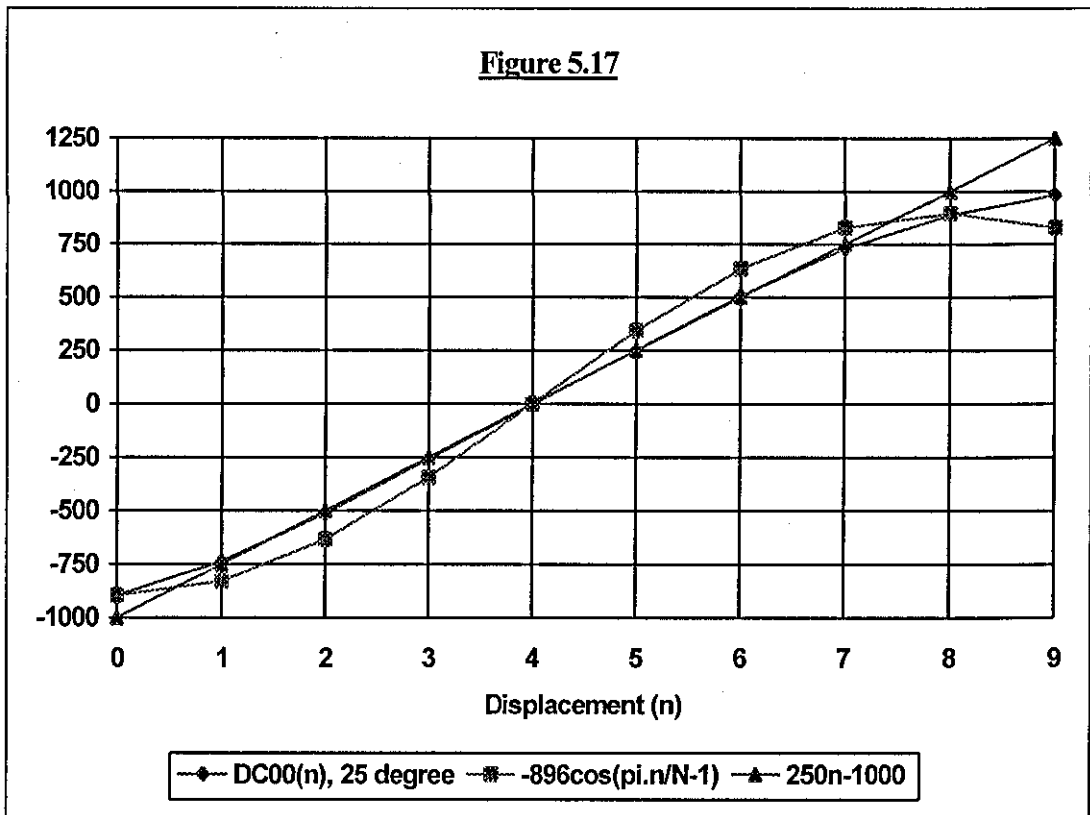


Figure 5.18 shows the modelled function for AC10(n) for a 25-degree edge. A similar result is obtained for AC10(n) for a 45 degree edge in Figure 5.19. The contrast difference is 255 in each case

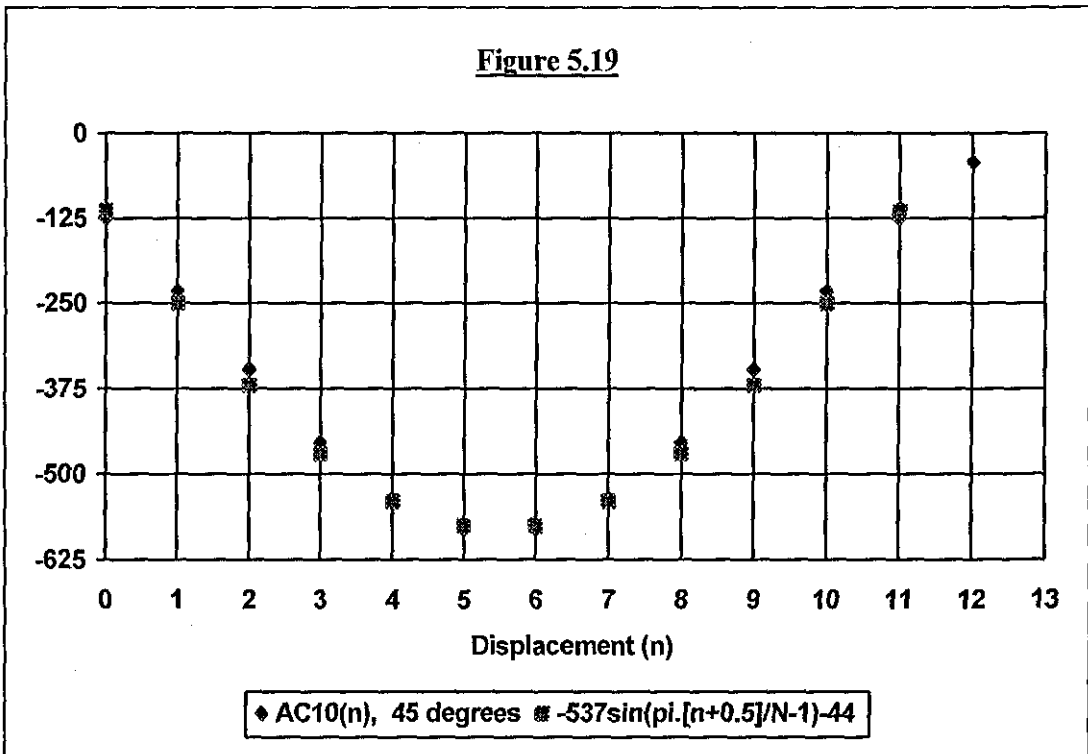
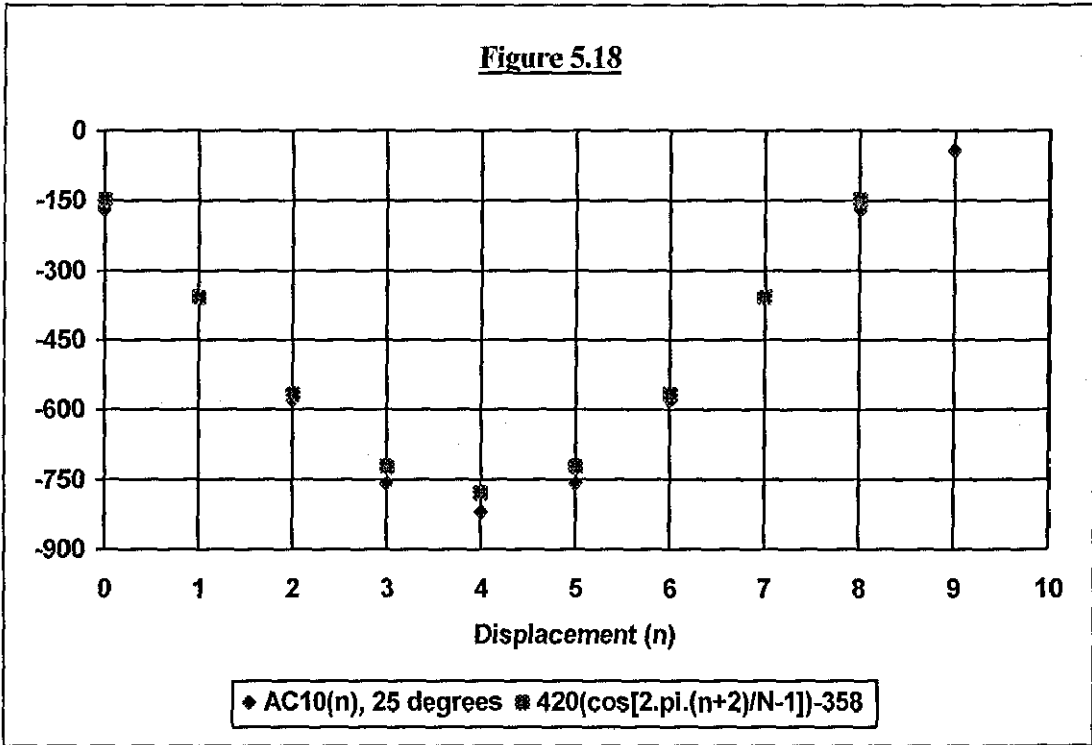
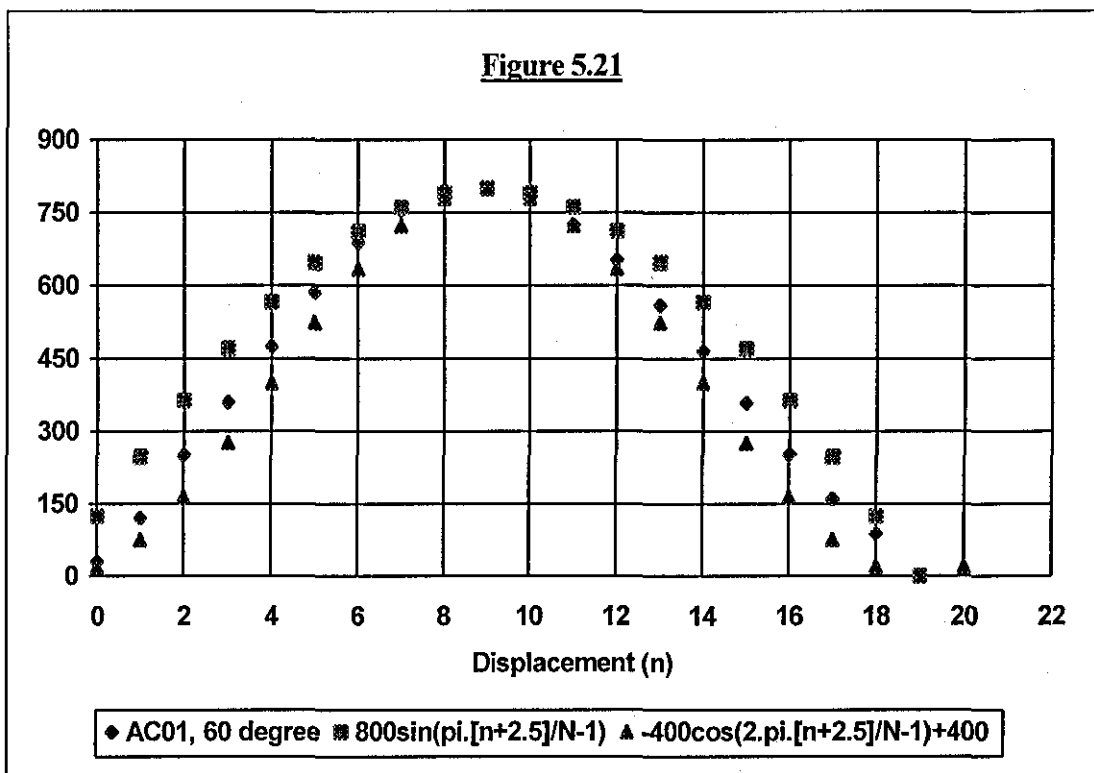
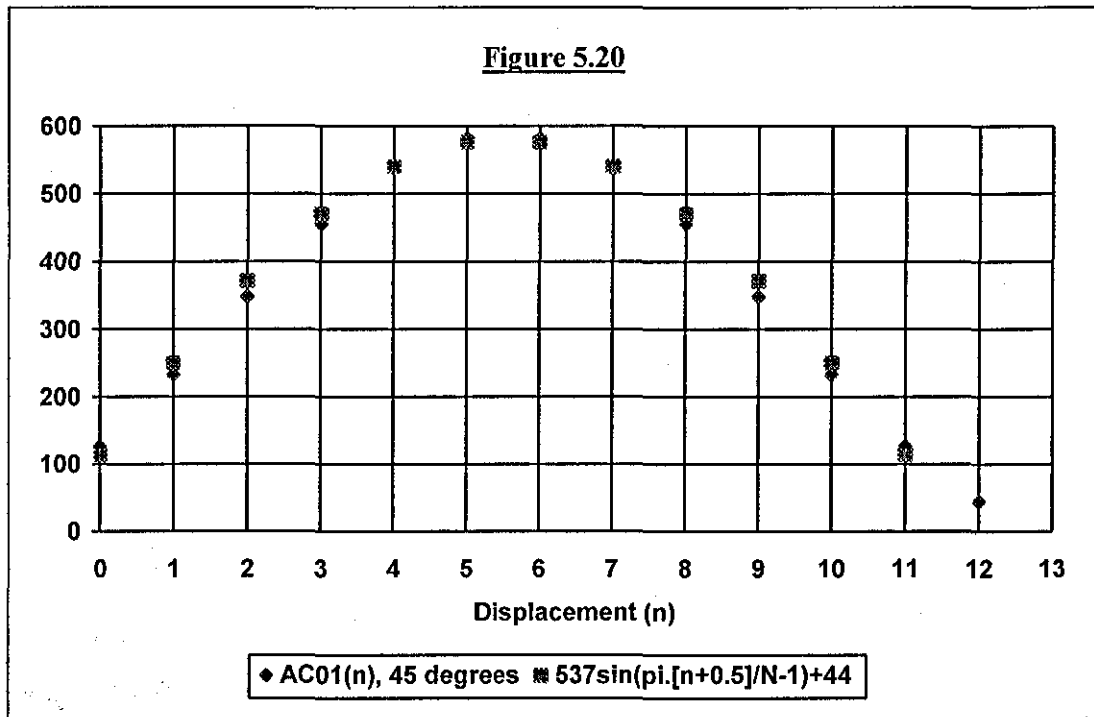


Figure 5.20 shows the modelled function for AC01(n) for a 45-degree edge. A similar result is obtained for AC01(n) for a 60 degree edge, Figure 5.21. A similar result is also obtained for 80 degree edges. The contrast difference is 255 in each case.



Figures 5.22, 5.23, 5.24 and 5.25 show the modelled functions for AC11(n) for 25, 45, 60 and 80 degree edges. The contrast difference is 255 in each case

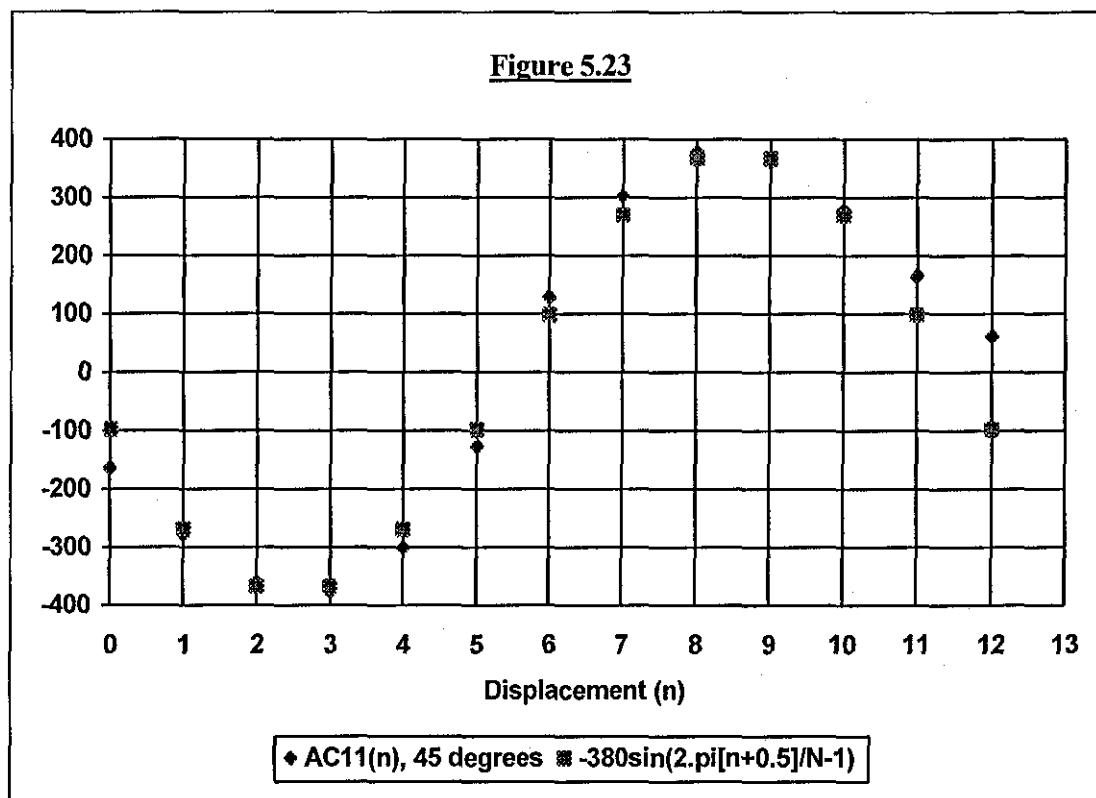
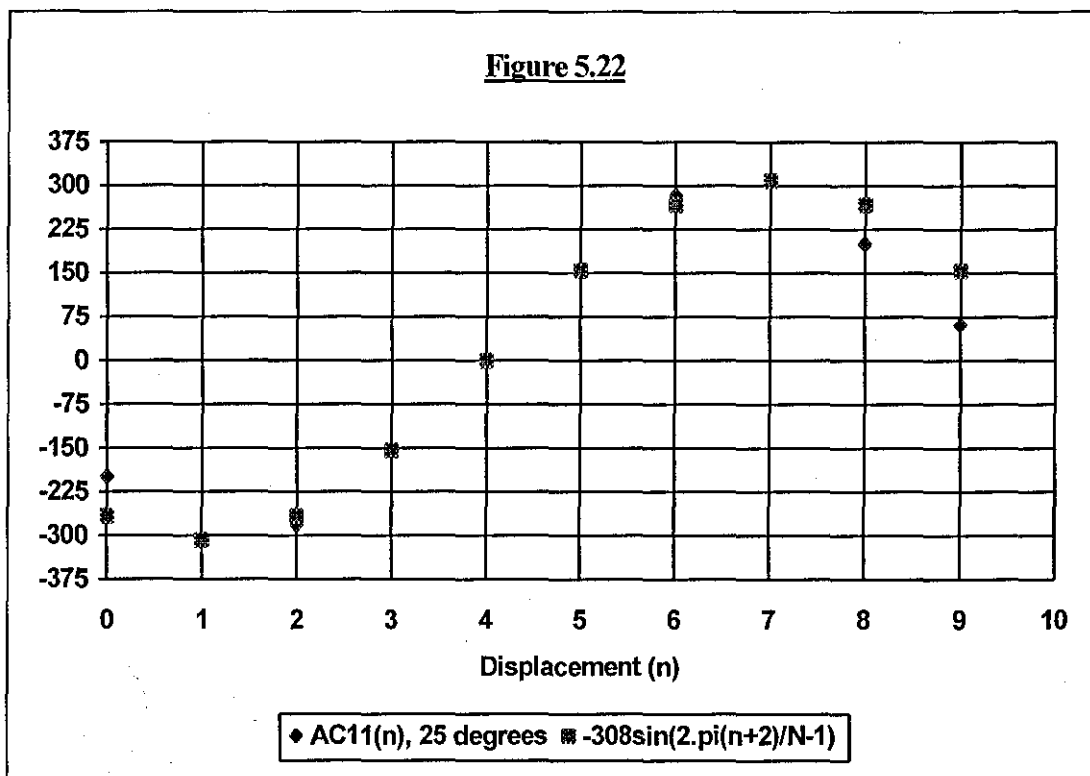
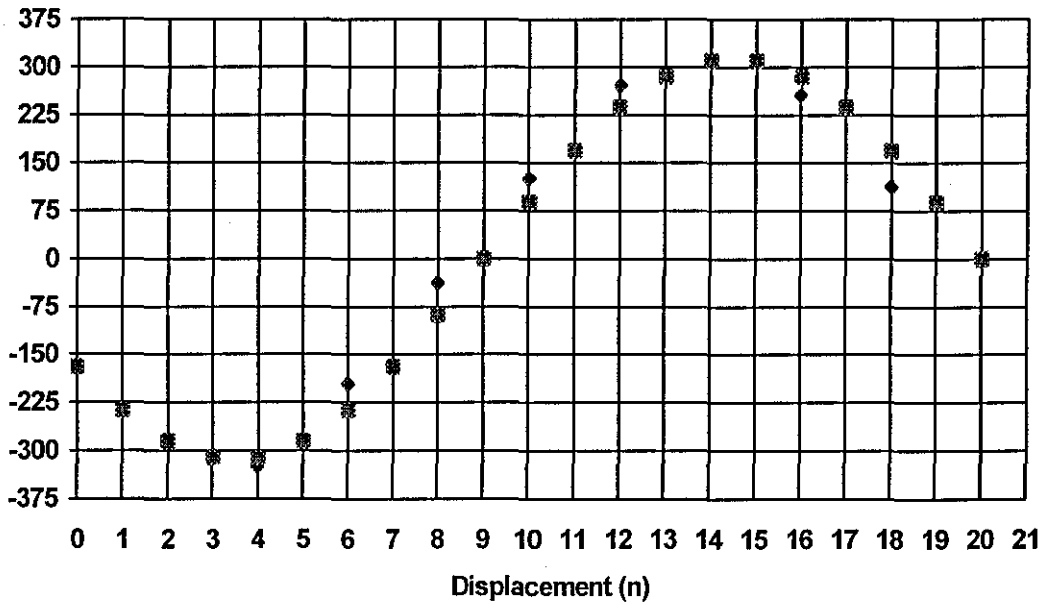
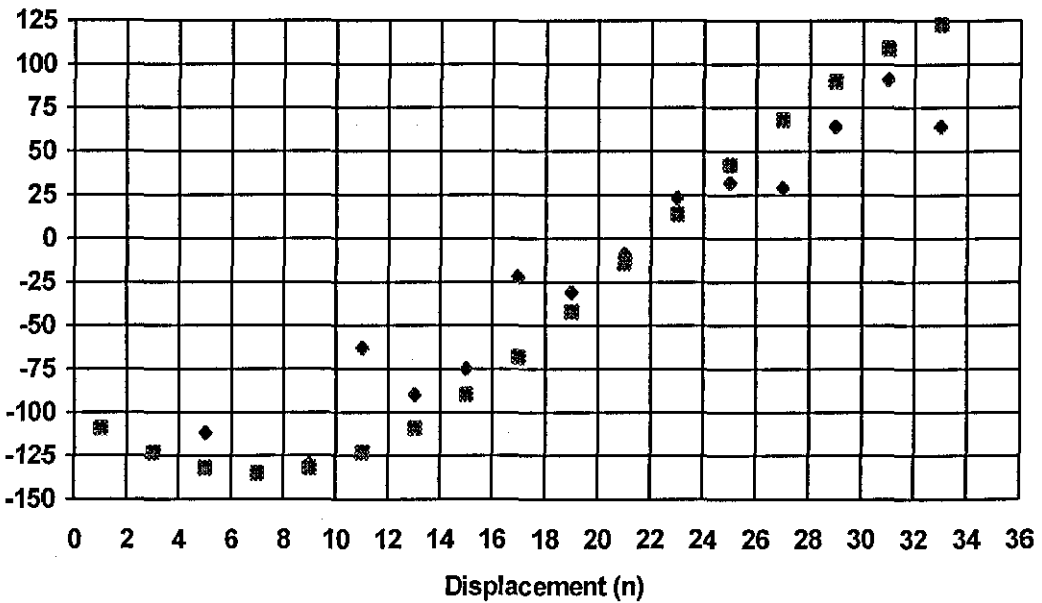


Figure 5.24



◆ AC11(n), 60 degrees ■ -313sin(2.pi(n+2)/N-1)

Figure 5.25



◆ AC11(n), 80 degrees ■ -135sin(2.pi(n+8)/N-1)

For AC01(n) for a 25 degree edge and AC10(n) for 60 and 80 degree edges, the characteristics for these coefficients are not easily modelled, due to their shape. However, for any given edge, if AC01(n) is not easily modelled, due to its shape, then AC10(n) will always be easily modelled, and vice versa.

The concealment process is split into two distinct sections i.e. (i) Inspection of the classification results to ascertain whether an edge is present in the surrounding blocks that will enter the error block, or not as the case may be. (ii) To calculate the missing coefficients for the error blocks, using such methods as decided by the previous section (i). There are three possibilities for the calculation of the transform domain coefficients. (1) If all the surrounding blocks contain Flat blocks or blocks having no significant edge/feature information, then the Linear Interpolation of the surrounding blocks coefficients is performed. (2) If one or more of the surrounding blocks contains an edge or edges but this edge does not enter the error block under observation, due to the edges angle and displacement, then a form of Linear Interpolation is employed. Specifically, only those blocks with a Flat classification are used to perform the Linear Interpolation of the coefficients. An example of this situation is shown in Figure 5.26 Edge 2, Error block L1. (3) One of the surrounding blocks contains an edge that will enter an error block, then the transform domain coefficients are calculated as explained in Section 5.1 following.

There are two assumptions made in the concealment process (i) The edge remains approximately straight through the surrounding and the error blocks i.e. the angle of the edge is approximately the same in these blocks. Although the results produced by the algorithm when there is a change of angle through the error and surrounding blocks are still significantly better than those produced by Linear Interpolation. (ii) The edge contrast remains approximately the same through these blocks.

The concealment process is the same irrespective of the classifier chosen to classify the surrounding blocks; a Flow Chart of the algorithm is shown in Figure 5.27. However, the concealment stage has to be aware of which classifier is being used i.e. the amount of diagonal classes within the classifier. The concealment algorithm then inspects the surrounding blocks looking for an edge classification that is consistent

with an edge appearing in the error block under observation. For example, using the D3 classifier all diagonal angles are classed into three possible regions, having central values of 25, 45 and 65 degrees.

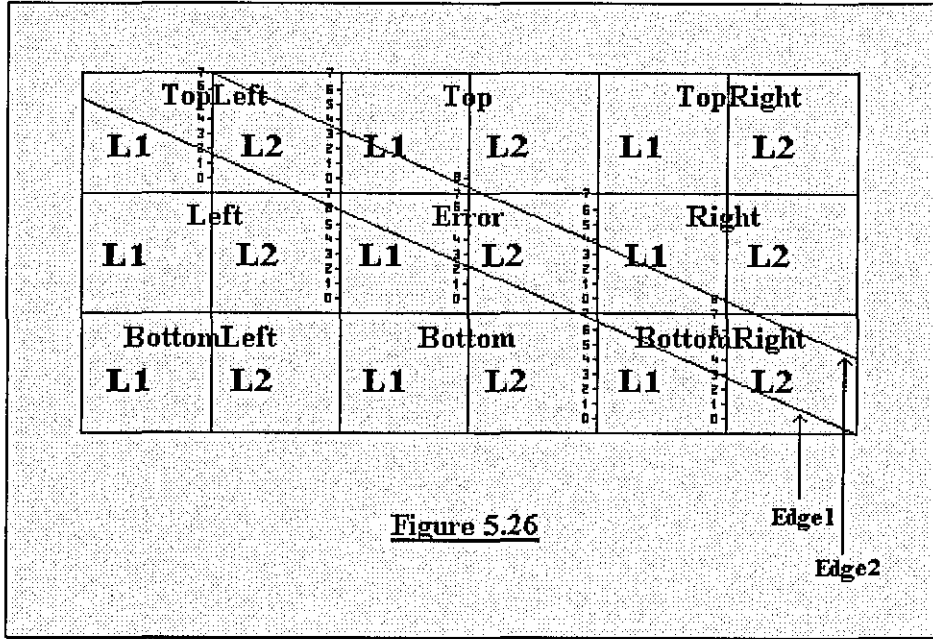


Figure 5.26 shows 3 by 3 MCU's (two Luminance blocks per MCU), the central MCU is in error due to a lost cell in an ATM network. As can be seen, two edges are present having an angle of 25 degrees in each case. For Error block L1 and L2 the concealment algorithm checks the classification results in the seven surrounding blocks, the eighth block being L2 or L1 of the error block. If Edge 1 is the only edge present, then after the classification stage TopLeft L2 will be classed as a block containing a 25 degree, down hill edge at a displacement of 2. Top L1 and L2: Flat block, Left L2: Flat block (no 25 degree edge of displacement of 10), BottomLeft L2: Flat block, Bottom L1 and L2: Flat block and BottomRight L1 will be classed as 25 degree down hill edge at a displacement of 6. If Edge 2 is the only edge present, the classifications will be: - TopLeft L2, 25 degrees down hill, displacement 7, Top L1, 25 degrees down hill, displacement 3, Top L2, Flat, Left L2, Flat, BottomLeft L2, Flat, Bottom L1 and L2, Flat, BottomRight L1, Flat, BottomRight L2, 25 degree down hill, displacement 8 and Right L1, 25 degree down hill, displacement 4.

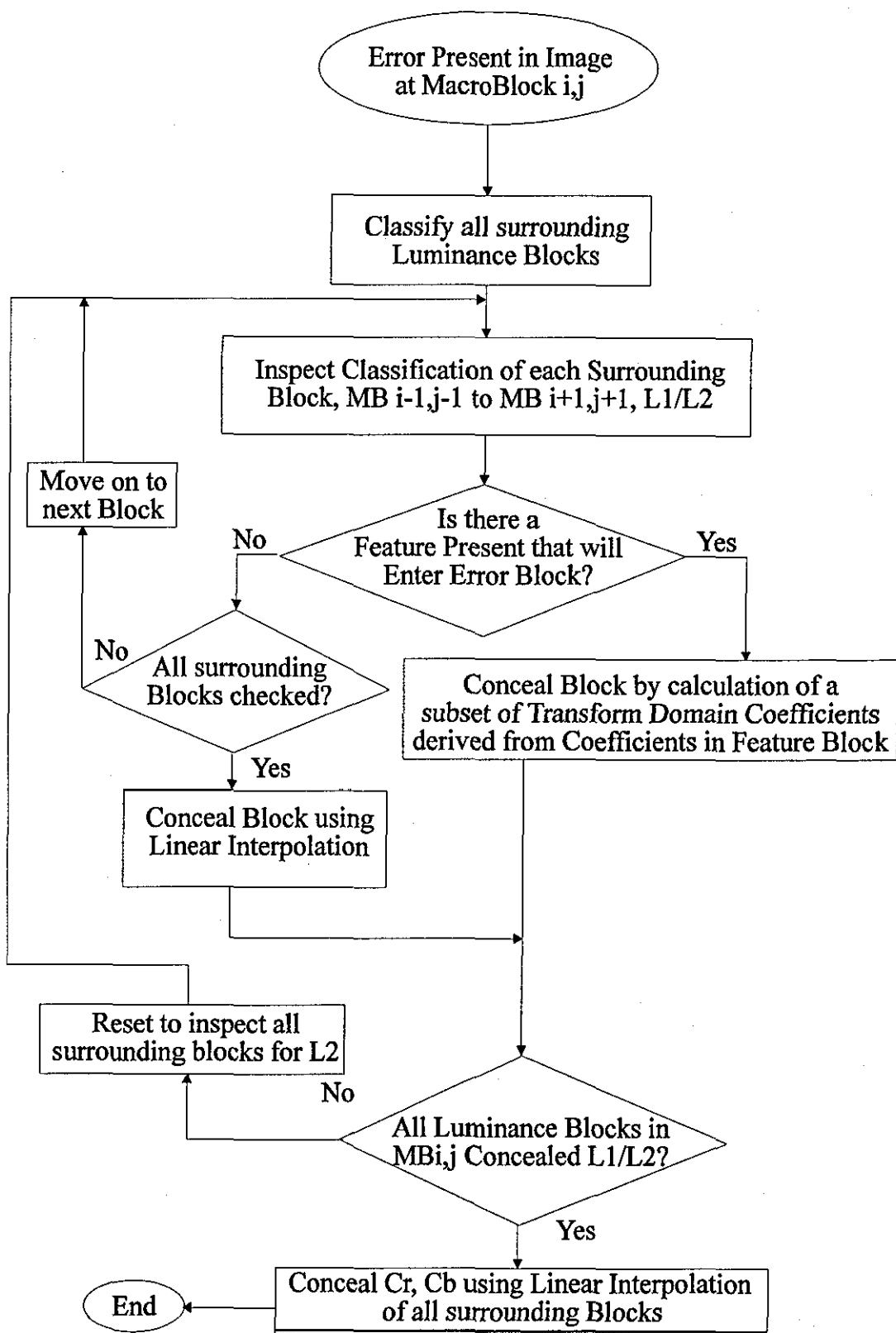


Figure 5.27

When the concealment algorithm detects an edge classification it checks to see if the edge will enter the error block in question. Using the D3 classifier, a 25 degree edge in TopLeft L2 will enter L1 of the error block if its displacement is less than 6. If the edge enters error block L1, the algorithm checks to see if the edge is present in Bottom L2 and or BottomRight L1, with a value of displacement which is approximately consistent with the edge in TopLeft L2. This confirms the fact that the edge will be present in error block L1. Therefore, if Edge 1 is the only edge present, the edge will enter error block L1. However, if Edge 2 is the only edge present the edge will not enter error block L1.

Using the Initial Classifier and Concealment algorithm [31], a number of incorrect concealments could be performed due to the following.

1. Classification of TopLeft L2 as a Flat block, due to the fact that Edge 1 just enters the block. Also, if the contrast on either side of the edge is similar a misclassification will result. This will cause the algorithm to conceal Error block L1 as a Flat block, producing a visible discontinuity in the edge.
2. The Initial Classifier does not assign a value of displacement or position to a diagonal edge classification, therefore when Edge 2 is the only edge present the algorithm has no way of knowing if this edge will enter Error block L1.
3. When the algorithm decides that an edge in the surrounding blocks enters the Error block in question, the concealment process Linear Interpolates the transform domain coefficients in the surrounding blocks containing the edge, which produces incorrect values for the coefficients as explained in the following sections.

As can be seen from Figure 5.26 an edge in the Error block L1 or L2 is displaced by a fixed amount from the edge in a surrounding block. The value of the change in displacement is dependent on the angle of the edge and the surrounding block containing the edge. This fixed shift in displacement is calculated for each class within each classifier and is made available to the concealment algorithm.

5.1 Transform Domain Coefficient Calculation.

Once the algorithm has detected an edge that will enter the error block, the next stage is the calculation of the following Luminance coefficients, $DC00(n)$, $AC01(n)$, $AC10(n)$ and $AC11(n)$.

5.1.1 $DC00(n)$.

The first coefficient to be calculated is the $DC00(n)$ coefficient and from Figures 5.1 to 5.4 and Figure 5.17 it can be seen that irrespective of angle and the difference in contrast on either side of the edge, a straight line function can be used to model this coefficient.

Now, from the surrounding blocks containing the edge, the angle and displacement for each block has been calculated/estimated by the classification algorithm. In addition, the value of $DC00(n)$ for the particular displacement (n) in each case and the value of the change in displacement ($n+/-x$) required in the error block, is also known. Therefore, using equation (5.0) and the above information, the value for M and C can be found using the simultaneous equation method. Once these two values are found, equation (5.0) can be used again to calculate the $DC00(n)$ value for the error block, for the particular value of displacement (n) required in the error block.

The question may be asked that if two values for $DC00(n)$ are known, why the average of the two is not used for the error block as is the case with the Linear Interpolation concealment method. The reason for this is that in most cases the value of displacement (n) of the error block is not centrally located between the displacements of the two surrounding blocks containing the edge. Moreover, the situation where the displacement of the two surrounding blocks may be situated at one end of the displacement scale and the displacement required in the error block is at the other end of the scale is a regular occurrence using a particular class within each classifier. Edges that enter the class nearest the vertical in each classifier often exhibit this characteristic.

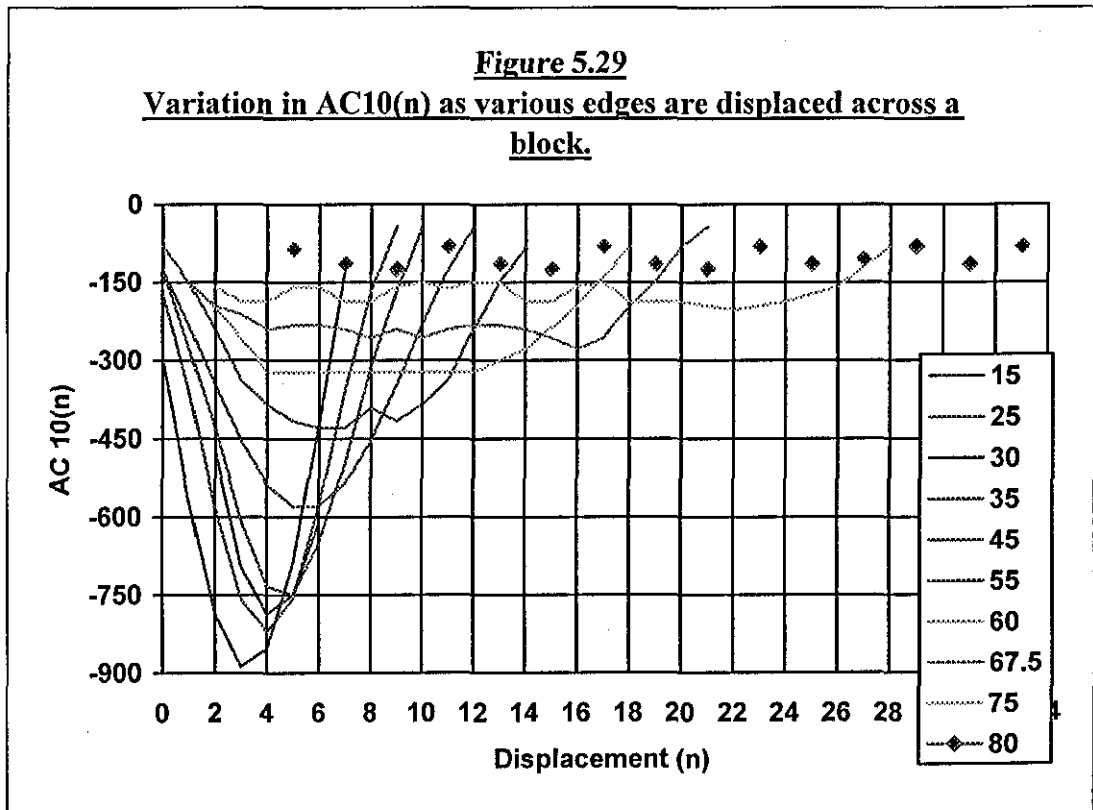
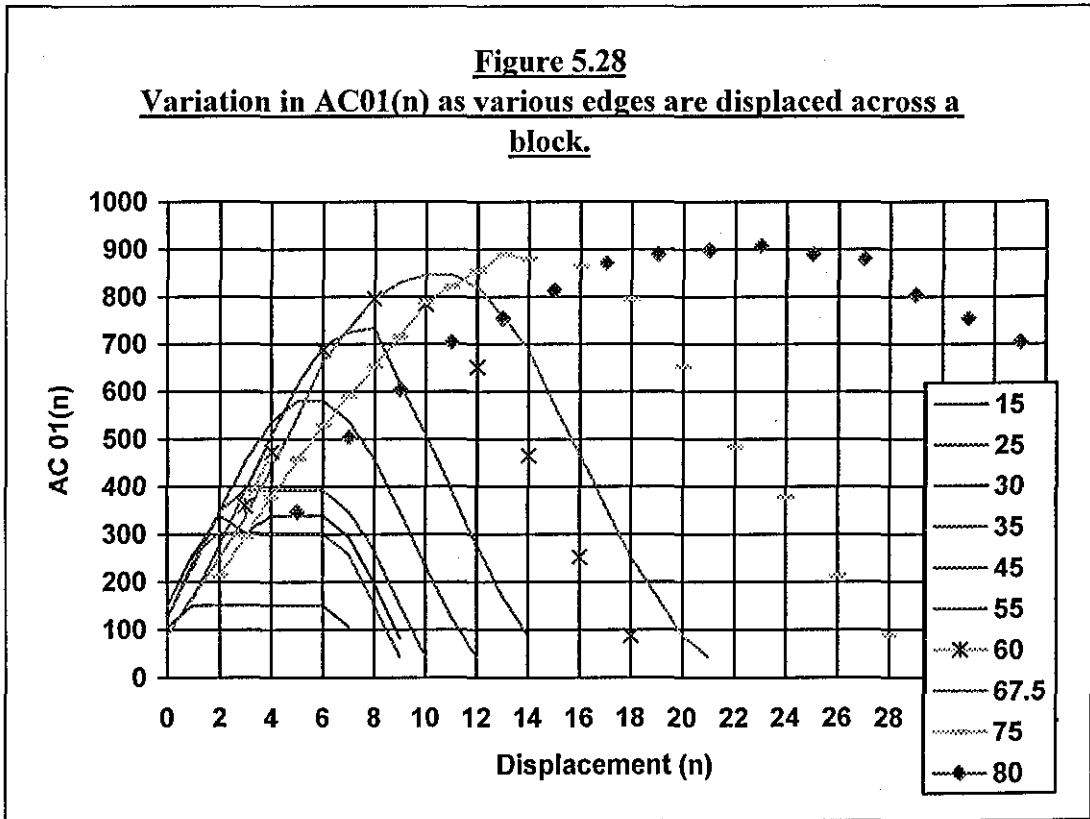
5.1.2 AC01(n) and AC10(n).

As shown in Chapter 4 the Ratios 1, 2 and 3 can be calculated using empirically modelled equations, or derived experimentally from idealised edges and stored in a look-up table. Depending on the classifier and the particular class within the classifier, different values for the Ratios 1, 2 and 3 are used. For example, using the D2 classifier there are two diagonal classes, as shown in Figure 4.24, the first includes all angles from 15 to 45 degrees, the second includes all angles from 45 to 75 degrees. If the classification algorithm has detected an edge in a surrounding block that falls within the angles of (i) the first class it is classed as a 30 degree edge, (ii) the second class it is classed as a 60 degree edge. Now, if the displacement of this edge is such that it will enter the error block in question the concealment algorithm uses the Ratio values for either a 30-degree edge, or a 60-degree edge.

Inspection of Ratio 2 and 3 in equations (4.0.2) and (4.0.3) shows that coefficient AC11(n) is common to both. Therefore, since the values for R2(n) and R3(n) are known, as explained above, if AC11(n) is calculated first for the error block then AC01(n) and AC10(n) can be simply calculated. This is achieved by one division in each case. This is one method for calculating coefficients AC01(n) and AC10(n) in the concealment algorithm and is called Method 1.

A second method for calculating coefficients AC01(n) and AC10(n) uses the fact that AC01(n) or AC10(n) can be empirically modelled as shown in equation (5.1), (5.3), (5.4), (5.6) and (5.8). Specifically, using the values for AC01(n) or AC10(n) in the two surrounding blocks containing the edge, the corresponding value of displacement (n), the equation for the angle classification, in conjunction with the simultaneous equation method, the unknown values X and Y can be calculated. Once these values are known, AC01(n) or AC10(n) can be calculated for the value of displacement (n) required in the error block. As explained earlier on page 18, the edge in most classes does not have easily modelled equations for both AC01(n) and AC10(n). For example, the 60-degree class in classifier D2 does not have an easily modelled AC10(n) coefficient, as shown in Figures 5.11, but does have an easily modelled AC01(n) coefficient, Figure 5.7. Interestingly, only the 45-degree class within the D3 classifier

has an easily modelled equation for both $AC01(n)$ and $AC10(n)$. This is confirmed in the graphs in Figures 5.28 and 5.29.



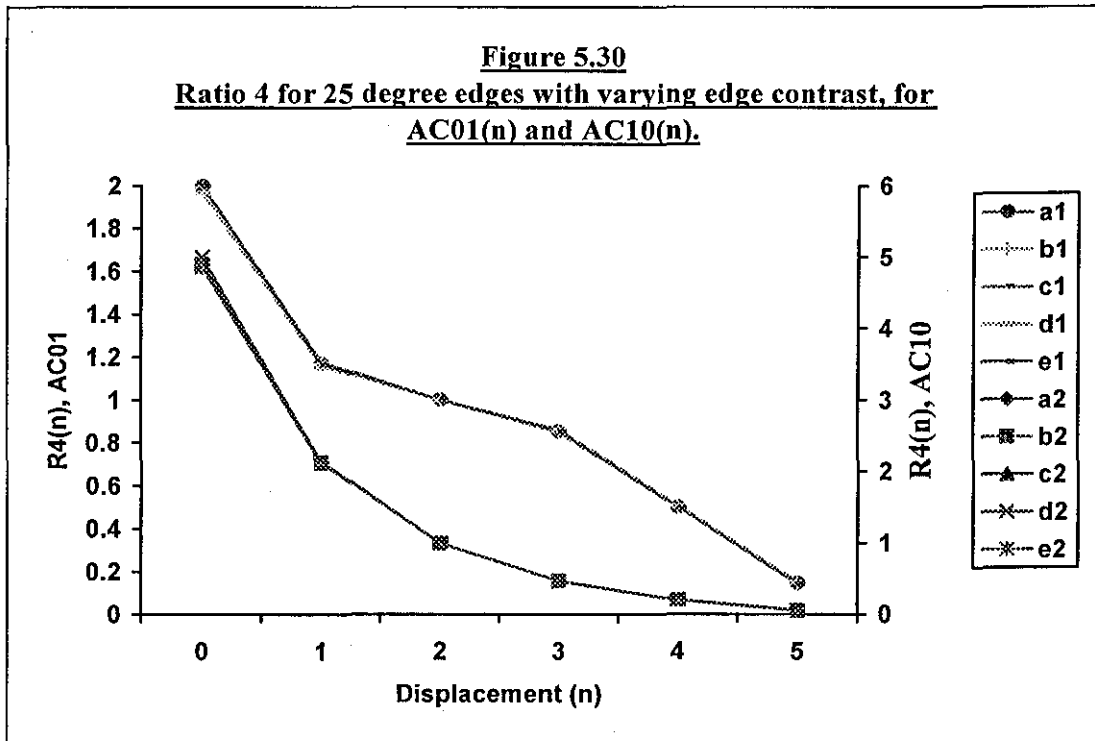
Therefore, once AC01(n) or AC10(n) is found the other coefficient can be calculated using equation (4.0.1) for R1(n). Alternatively, the unknown coefficient could be found by using equation (4.0.2) or (4.0.3) and the value for AC11(n), which would have to be calculated first. This is called Method 2.

A third method for calculating AC01(n) and AC10(n) again uses the predictable nature of the two coefficients. As can be seen from Figures 5.5 to 5.12 the shape of the characteristics for either coefficient is the same irrespective of the edge contrast. Also, the ratio of one coefficient at position (n) to another at position (n+x), is the same irrespective of edge contrast. Figure 5.30 shows this graphically. The range of values for the displacement (n), for an edge in a surrounding block, which results in the edge entering the error block, is known for each class and classifier. Also, the change in (n) to (n+x) or (n-x), required in the error block, is also known. These values were obtained experimentally by producing all possible edges in each surrounding block and noting the values of displacement that resulted in the edge entering the error block and the value of change (x) from the displacement (n) in the surrounding block. Also, values of displacement in the surrounding blocks that resulted in the edge missing the error block were noted. This was done for edges of all classes in each classifier. This ratio of (n+x) to (n) is called Ratio4 and is calculated as follows :-

$$R4(n) = AC01(n+/-x) / AC01(n). \quad \dots\dots (5.10)$$

$$R4(n) = AC10(n+/-x) / AC10(n). \quad \dots\dots (5.11)$$

Once all values are calculated, a lookup table can be used to calculate the unknown value of AC01/10(n+/-x) by multiplication of the R4(n) value and AC01/10(n) of the surrounding block.



The above graph shows five example edges having the same angle but different edge contrast. a1 to e1 are the same edges as a2 to e2, however, a1-e1 are used to demonstrate R4(n) for coefficient AC01(n) and a2-e2 for coefficient AC10(n) for a 25-degree edge.

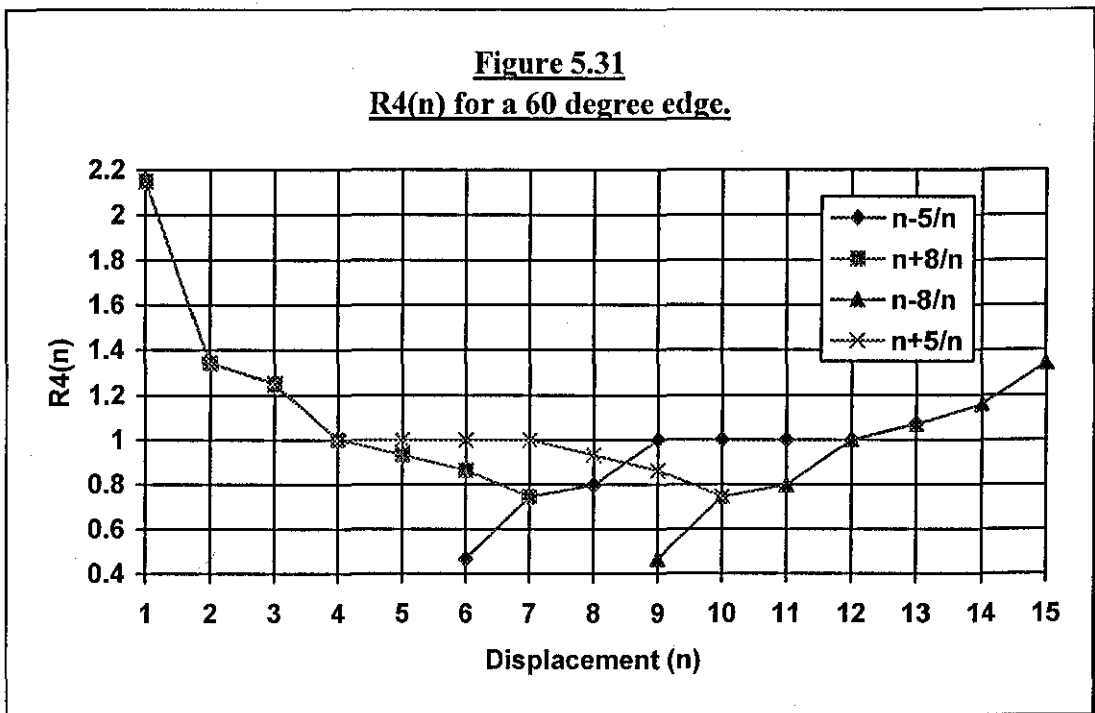


Figure 5.31 shows $R4(n)$ for a 60-degree edge in a surrounding block. As can be seen the value of change (x) required in the error block is different depending on the actual surrounding block containing the edge.

As for Method 2, once $AC01(n)$ or $AC10(n)$ is found the other coefficient can be calculated using equation (4.0.1) for $R1(n)$. Alternatively, the unknown coefficient could be found by using equation (4.0.2) or (4.0.3) and the value for $AC11(n)$, which would have to be calculated first. This method of concealment is called Method 3 and is preferred to the first two methods for the following reasons. Firstly, Method 1 is dependant on the accuracy of the calculated value of coefficient $AC11(n)$, therefore an incorrect or poor calculation of $AC11(n)$ would affect coefficients $AC01(n)$ and $AC10(n)$ producing a poor concealment. Method 2 uses transform domain information from the two surrounding blocks containing the edge and as can be seen from Figure 5.26, Edge 1, the two blocks are TopLeft L2 and BottomRight L1. Block TopLeft L2 is directly connected to the error block L1, or the next column of blocks, whereas BottomRight L1 is not directly connected to the error block L1, being in the next but one column of blocks. If the error block is in column (x) then TopLeft is in column ($x-1$) and BottomRight L1 is in column ($x+2$). Now, as shown in [xy] the correlation between blocks decreases as the separation between the blocks increases, which is conceptually obvious. Therefore, the edge in BottomRight L1 can induce inaccuracies into the calculation of coefficient $AC01(n)$ or $AC10(n)$ using Method 2, for these reasons. Another disadvantage of Method 2 is that the processing load and hence processing time is greater than either of the other two methods i.e. 4 subtractions, 4 additions, 4 multiplication's, 4 divisions and 3 Sine calculations. Method 1 requires one division and Method 3 one multiplication, for each coefficient $AC01(n)$ and $AC10(n)$. Also, Method 3 only uses the transform domain information in the surrounding block containing the edge that is directly connected or in column ($x\pm 1$) with respect to the error block. This fact becomes increasingly relevant as the size of an MCU increases in size to contain four Luminance blocks, 16 by 16 pixels. This is the standard size of a MacroBlock used within the MPEG standard.

5.1.3 AC11(n).

As shown in Figures 5.13 to 5.16 the peak value X for $AC11(n)$ is different for different values of edge contrast. Therefore the peak value for the particular contrast found in the edge of the surrounding blocks is required. The classifier has determined the angle and displacement of the edge in the surrounding blocks and with the value of $AC11(n)$ for these blocks the peak value can be calculated using either equation (5.2), (5.5), (5.7) or (5.9) or the equation for the particular class. Once the peak value is found the algorithm can simply calculate $AC11(n)$ at the required displacement ($n \pm x$) for the block in error using the chosen equation, this is called Method 1 and requires 2 sine calculations, 2 additions, 3 multiplication's and 3 divisions. Alternatively, if $AC01(n)$ or $AC10(n)$ is calculated first as described in section 5.1.2 then $AC11(n)$ could be calculated using equations (4.0.2) or (4.0.3) for $R2(n)$ and $R3(n)$ respectively. This is called Method 2 and requires 1 multiplication if $R2(n)$ or $R3(n)$ are held in a lookup table.

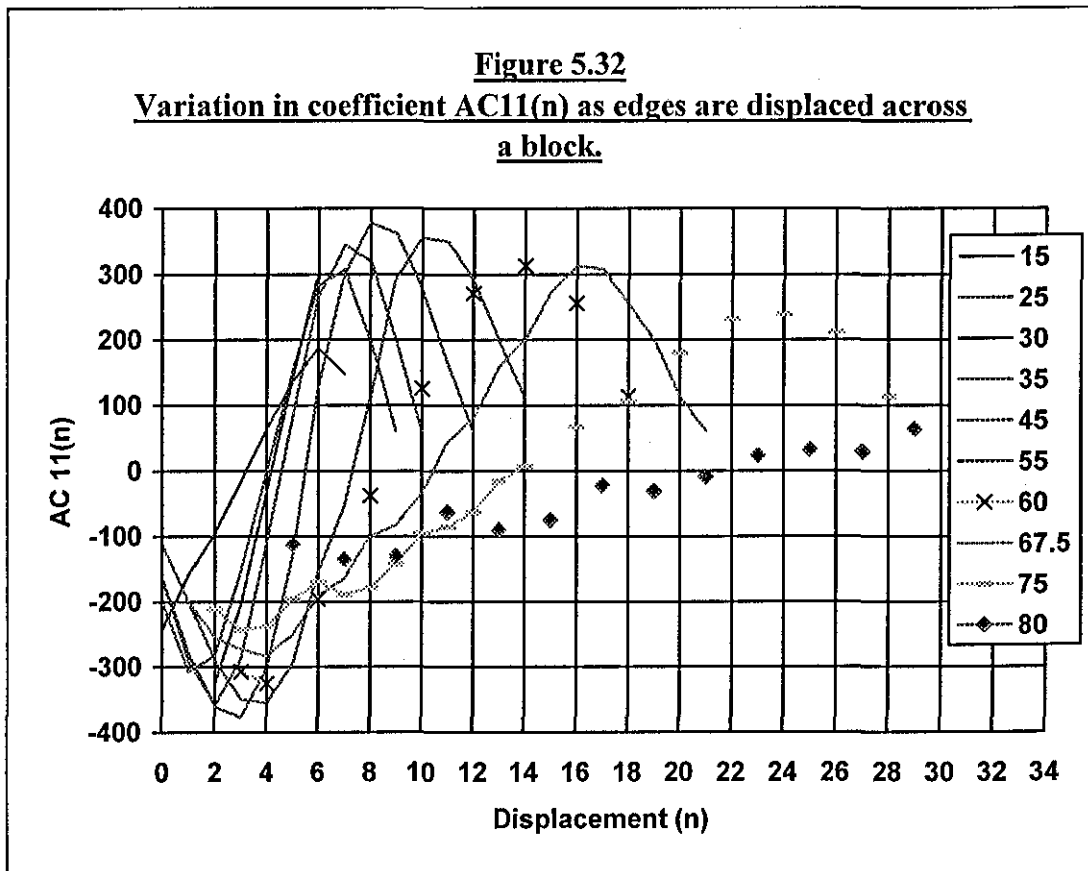


Figure 5.32 shows how coefficient $AC11(n)$ varies as different angled edges are displaced across a block. As can be seen the shape of the graph is approximately a sine function irrespective of the angle of the edge.

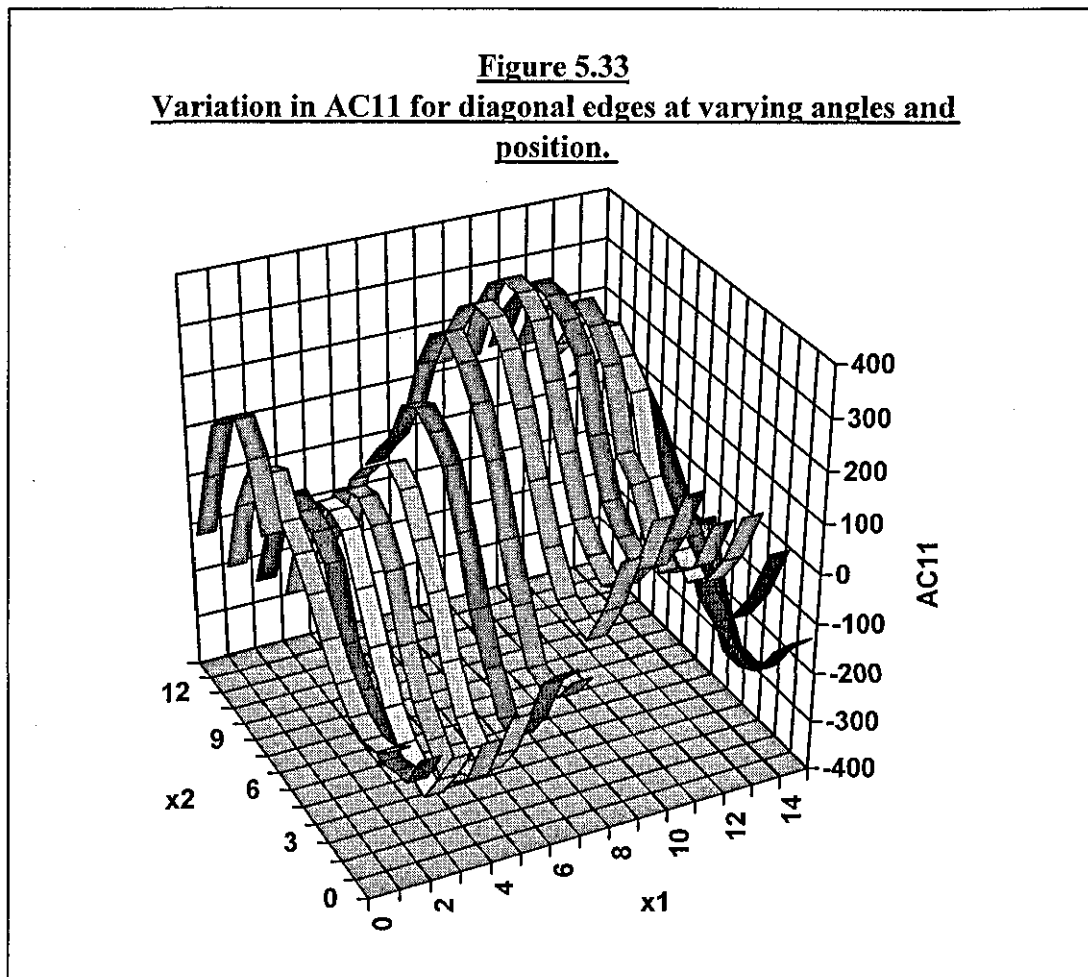


Figure 5.33 shows how coefficient $AC11$ varies as the angle and displacement of an edge varies. Alternatively, as explained in Chapter 4, the varying value of $AC11$ as $x1$ and $x2$ are varied. Each value of $x1$ corresponds to a value of displacement, and if a single value is chosen then as $x2$ varies the angle of the edge with the given value of displacement changes. This information could be used in situations where the classification of edges in the surrounding blocks to an error block have different values for the angle of the edge, as opposed to a difference in the value of displacement only. In such an example not only the displacement of the edge in the

error block is changed with respect to the surrounding blocks, but the angle of the edge also changes in the concealed block. This situation could occur when an error is present in a curved or semi-circular edge, as long as the radius of the edge is large enough. If the radius of the edge is below a certain value, then when the block is classified the classifier would produce a classification of Unknown. This would cause the concealment algorithm to conceal the error using the Intelligent Linear Interpolation technique. If the radius is large enough the edge would not cause a misclassification from the classifier, and the concealment can be performed. This would include the calculation of the other three Transform Domain coefficients in addition to coefficient AC11. Interestingly, the value of AC11 can be modelled as a sine wave in both the x_1 and x_2 axis. This would produce simple equations for the calculation of AC11 in such situations.

5.1.4 All Remaining Coefficients.

All the remaining Transform Domain Coefficients are set to zero, although as shown in Figures 5.34 to 5.42 further coefficients could be easily calculated.

All the coefficients within the chrominance blocks are concealed by Linear Interpolation of the chrominance in the surrounding blocks irrespective of the presence of any features. This technique does not significantly affect the colour image on average as most high-frequency components of a colour image are primarily in the Luminance component [36].

As shown in Chapter 4.2, the significant coefficients when an edge is present in an 8x8-pixel block are AC01, AC10, AC11, AC12, AC21 and AC22. These coefficients are usually still present after the quantisation of the coefficients in the encoding process. Coefficients AC02, AC20, AC12, AC21 and AC22 are shown in Figures 5.34 to 5.42, for 25 degree and 60 degree edges. Again, the shapes of the graphs are the same for all five examples of edge contrast. Therefore, these coefficients could be calculated by either finding the empirically modelled equations, or by calculating the Ratio 4 for each particular coefficient.

Figure 5.34
Variation in AC02(n) for 25 degree edge.

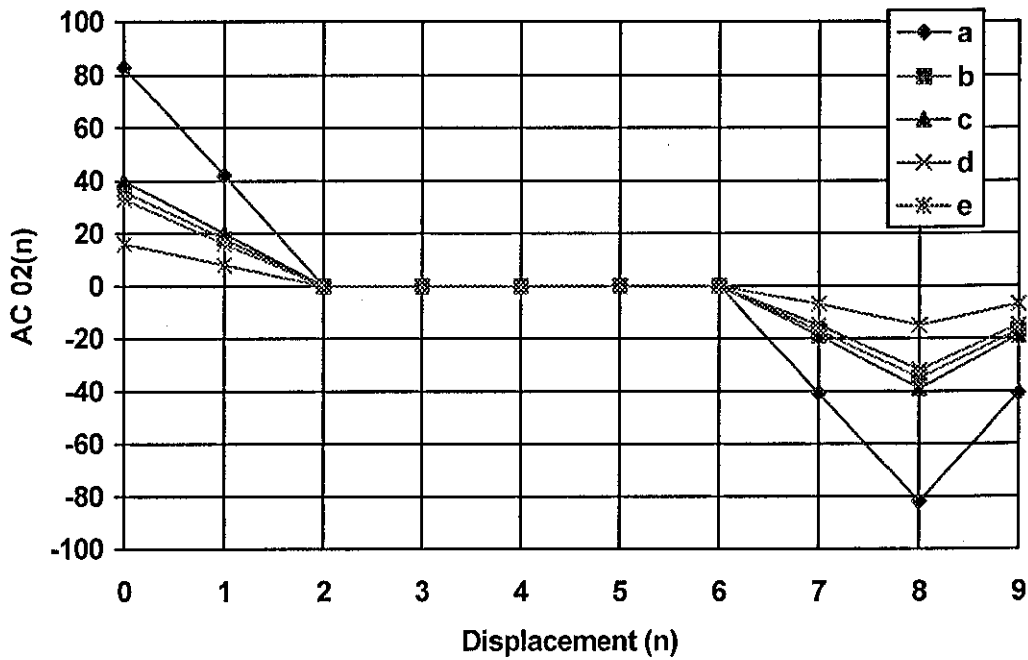


Figure 5.35
Variation in AC02(n) for 60 degree edge.

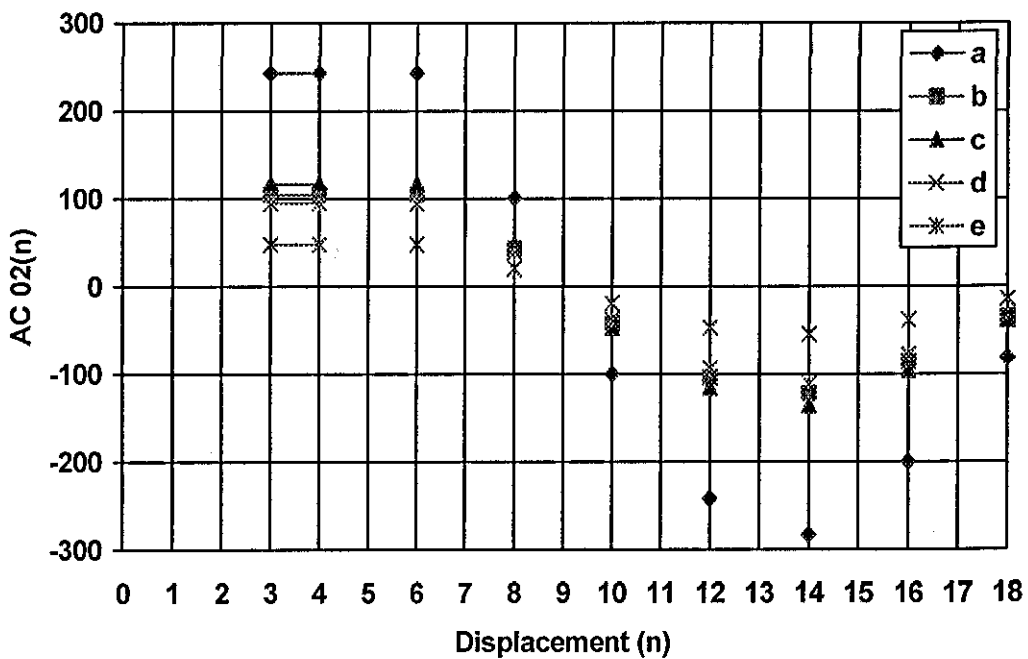


Figure 5.36
Variation in AC20(n) for 25 degree edge.

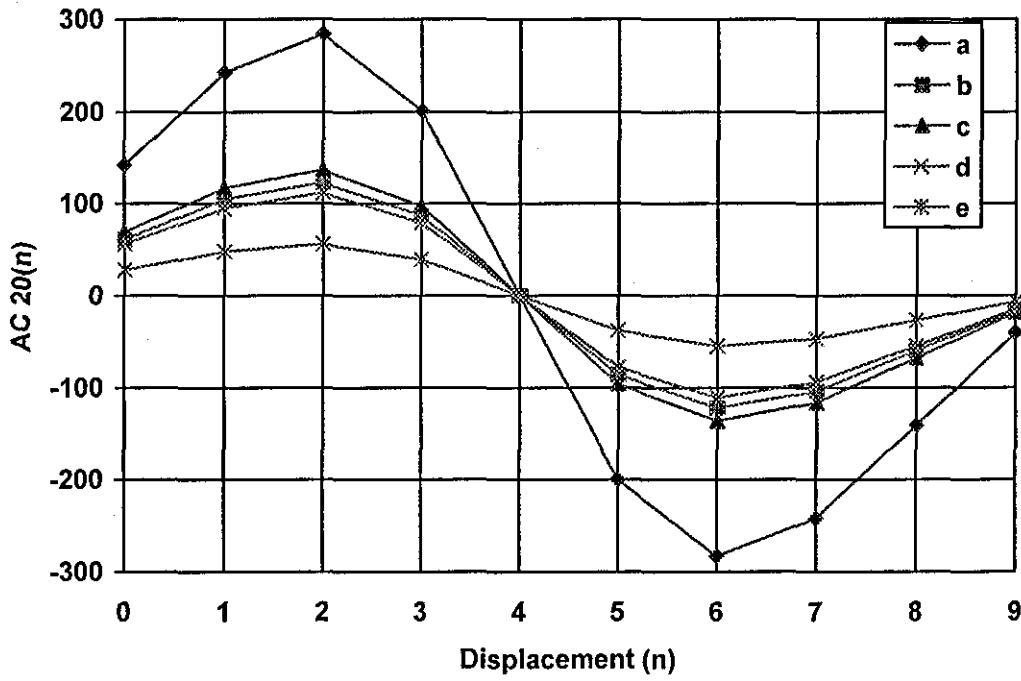


Figure 5.37 Variation in AC20(n) for 60 degree edge.

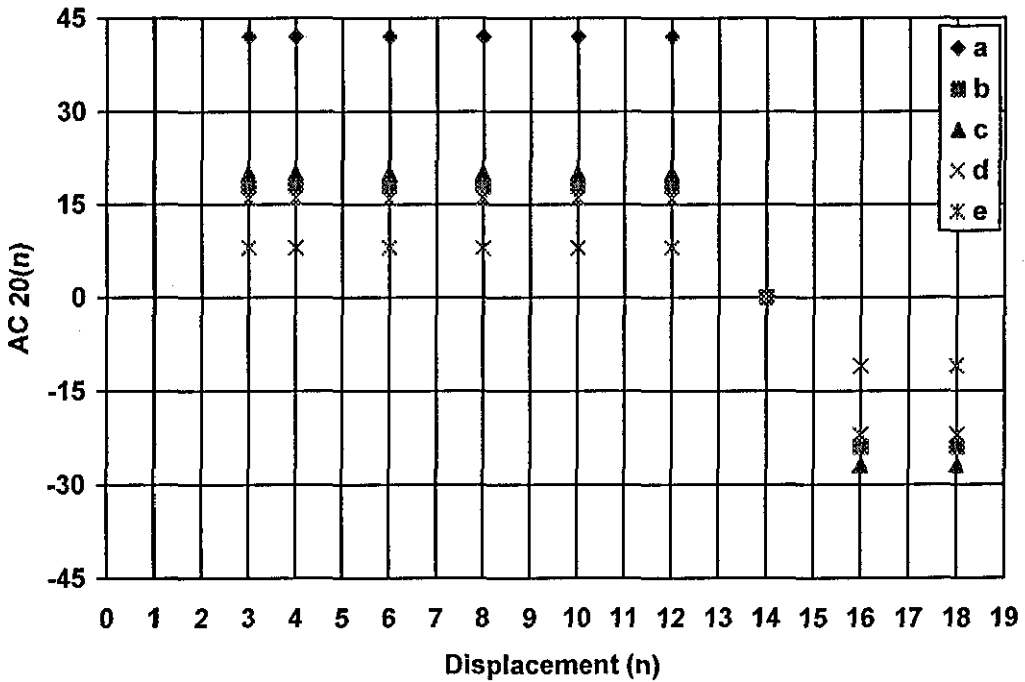


Figure 5.38
Variation in AC12(n) for 25 degree edge.

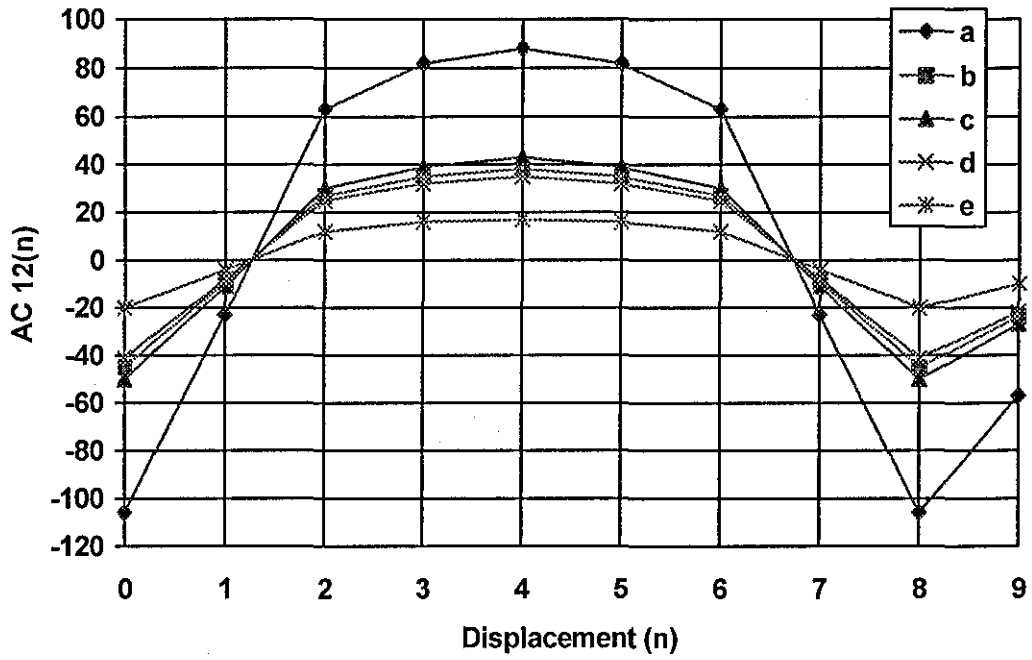


Figure 5.39
Variation in AC12(n) for 60 degree edge.

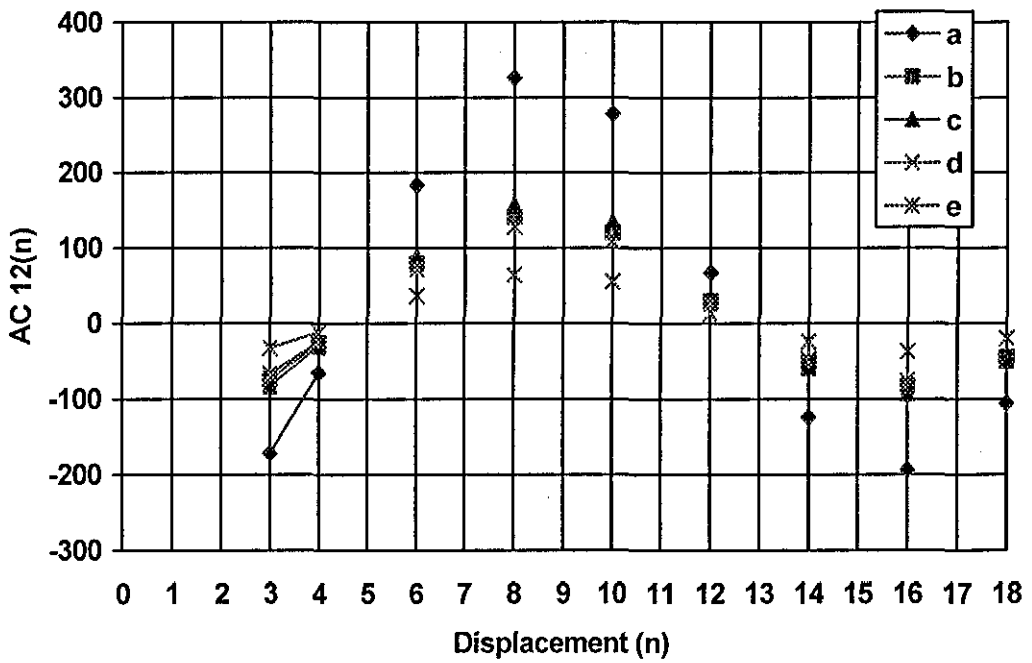


Figure 5.40
Variation in AC21(n) for 25 degree edge.

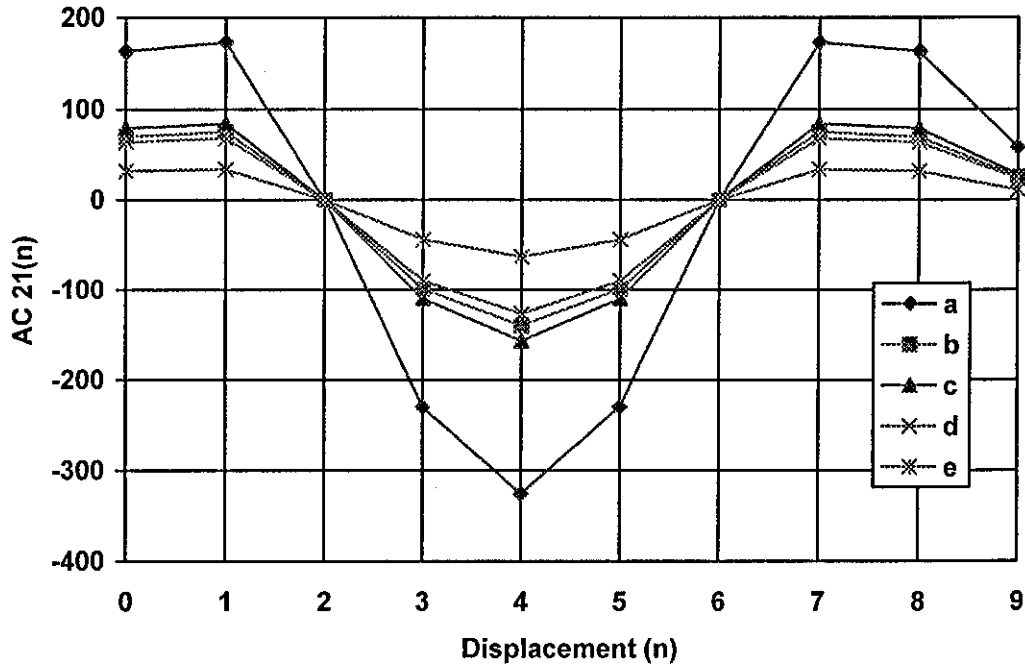
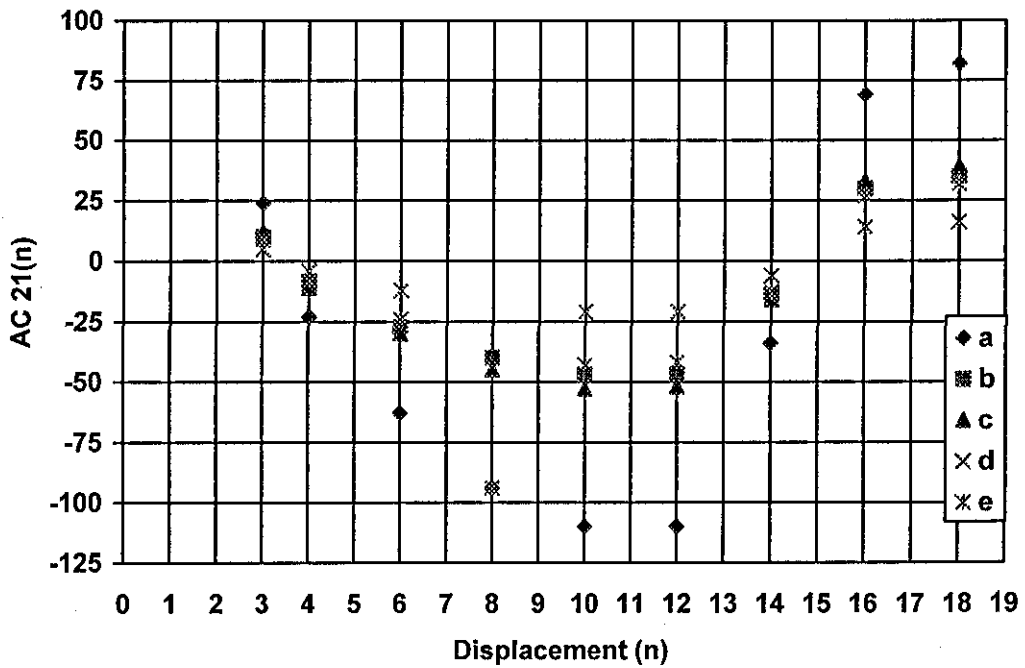
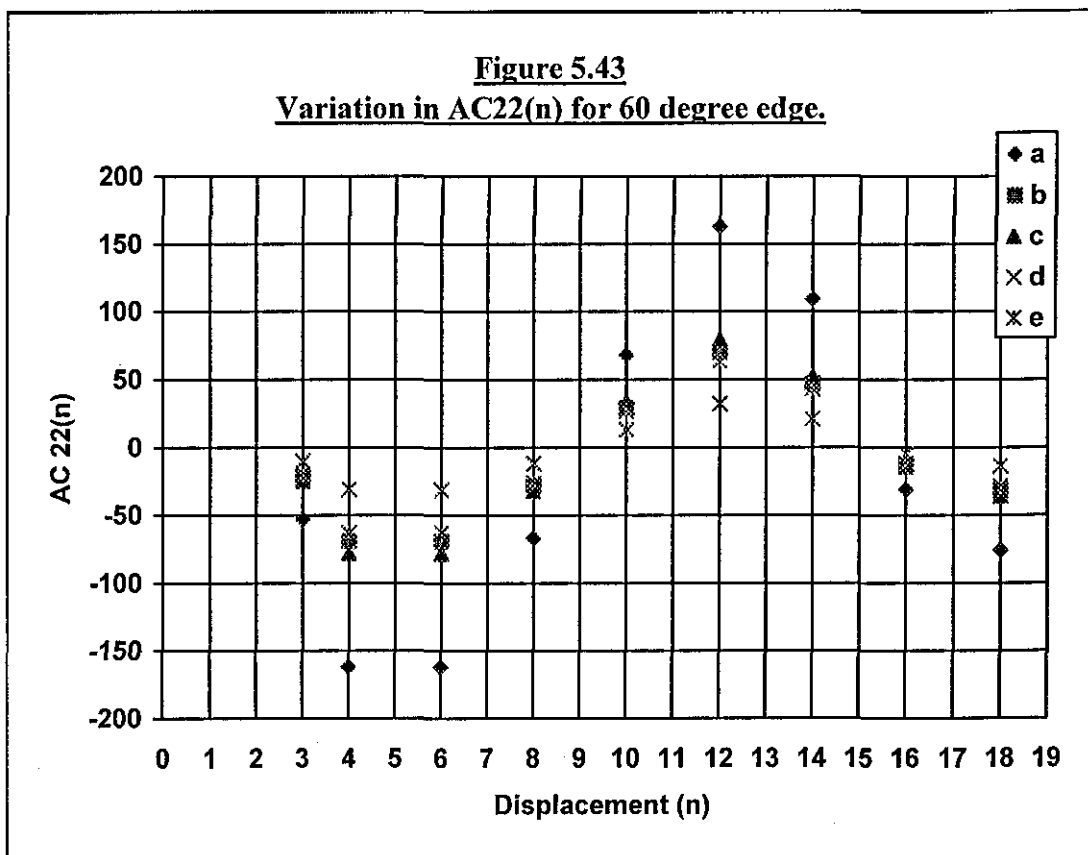
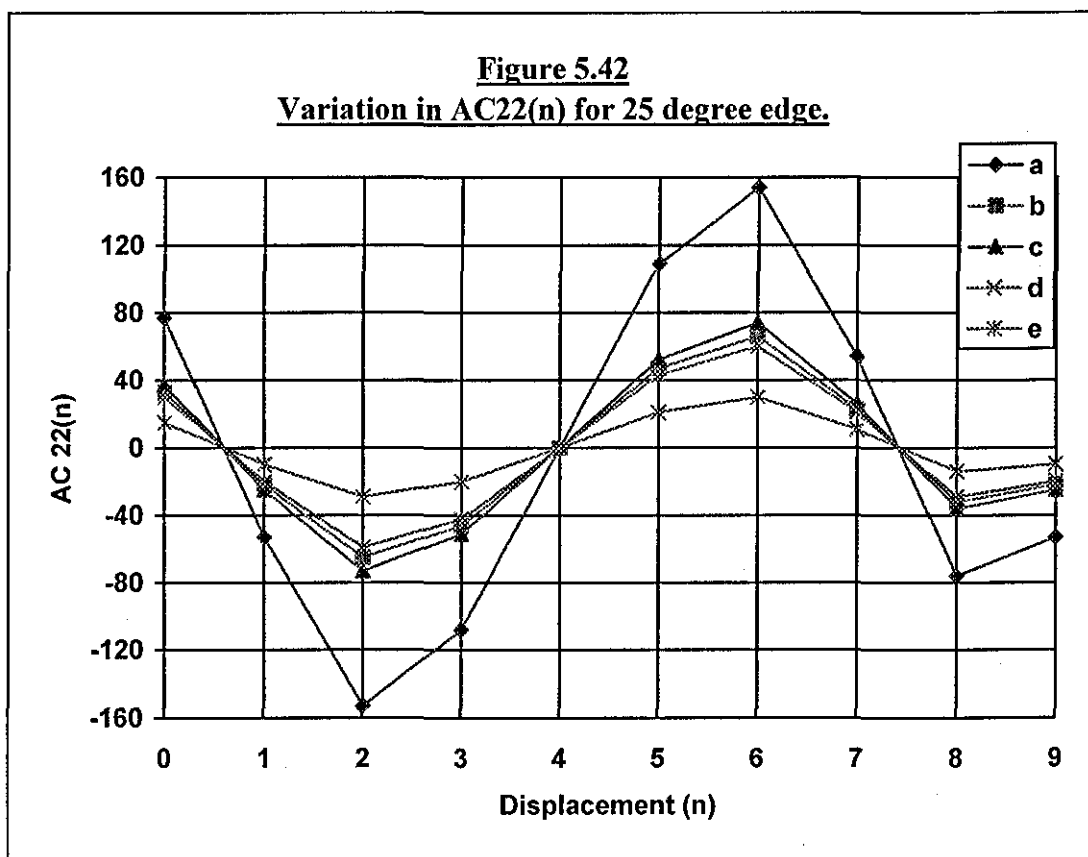


Figure 5.41
Variation in AC21(n) for 60 degree edge.





5.2 Results and Performance Evaluation.

The concealment algorithm can calculate the four coefficients by using different combinations of the methods for calculating $AC01(n)$, $AC10(n)$ and $AC11(n)$, each combination having a particular processing load and performance. Interestingly, the combination that requires the minimum amount of processing i.e. $DC00(n)$ algorithm, Method 3 for $AC01(n)$, $AC10(n)$, and Method 2 for $AC11(n)$, produces the best subjective results, Figure 5.47a. The processing requirements being 1 addition, 4 subtractions and 3 multiplication's. The improvement of one combination over any other combination is minimal having a maximum value of 0.192dB in some cases and being 0dB in other cases, using the PSNR objective measure for quality. However, the subjective performance appears to be better, using the minimum processing load algorithm, as seen in Figures 5.47a and 5.47b. For these reasons the concealment algorithm used to produce the following images use the above combination of methods. Significantly, the amount of processing required is not directly related to the size of the image sub-blocks used within the compression algorithm. Figure 5.44 shows some results of the detection, classification and concealment algorithms in use. Figures 5.44 and 5.46 have four separate images which are:- (i) The top left image is the original image with no errors introduced. (ii) Top right is the image with randomly distributed errors represented as black blocks. (iii) Bottom left, the errors have been concealed using the Initial Feature Detection, Classification and Concealment algorithm from reference [31]. (iv) Bottom right, the errors have been concealed using the Advanced Feature Detection, Classification and Concealment algorithm presented in this thesis, using classifier D4. As can be seen in Figure 5.44 there are errors where the Initial Detection, Classification and Concealment method has concealed using the Linear Interpolation of all the surrounding blocks. However, the advanced algorithm has only used those surrounding blocks that do not contain edge information. Figure 5.45 shows an example of this situation. Each error contains two Luminance blocks L1 and L2, and L1 of the top right error in Figure 5.45 is surrounded by blocks that have some form of edge information. In such a situation, the advanced algorithm has no choice but to use these blocks in the Linear Interpolation process. This explains why the concealment of L1 of the top right error has the same result in the Initial Concealment and Advanced Concealment algorithm.

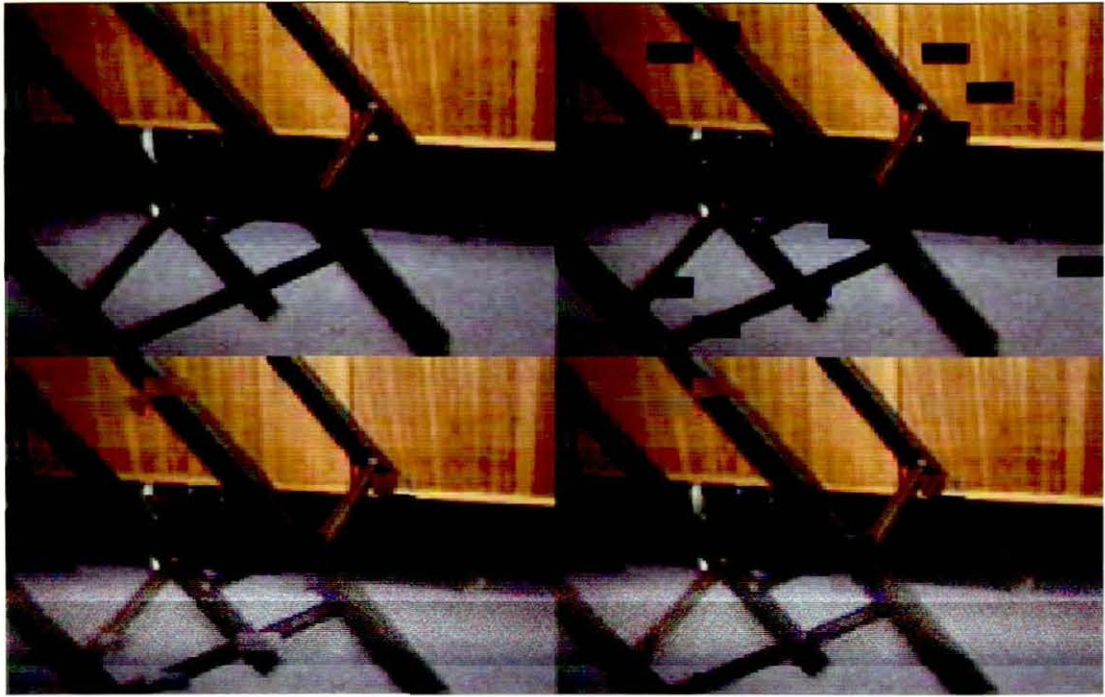


Figure 5.44

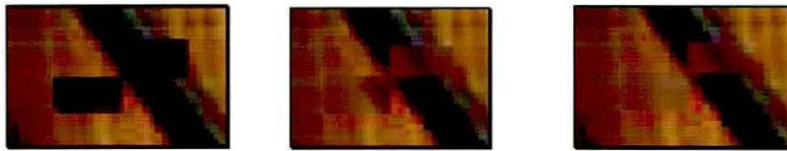


Figure 5.45

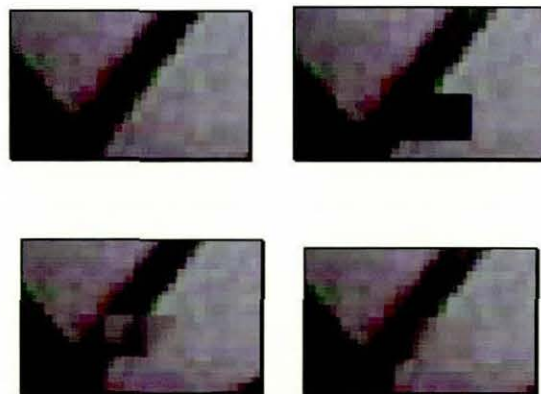


Figure 5.46

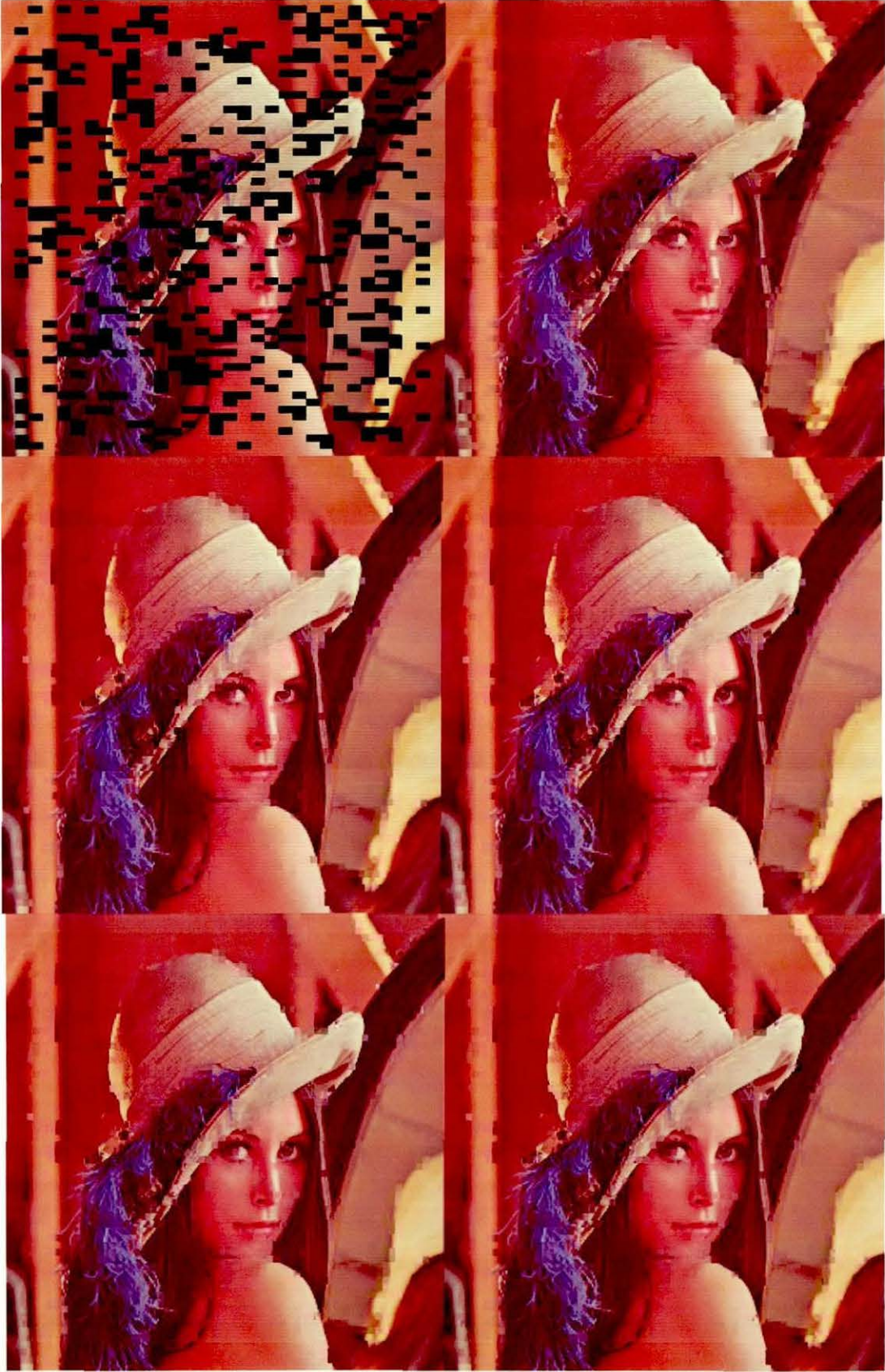


Figure 5.47a



Figure 5.47b

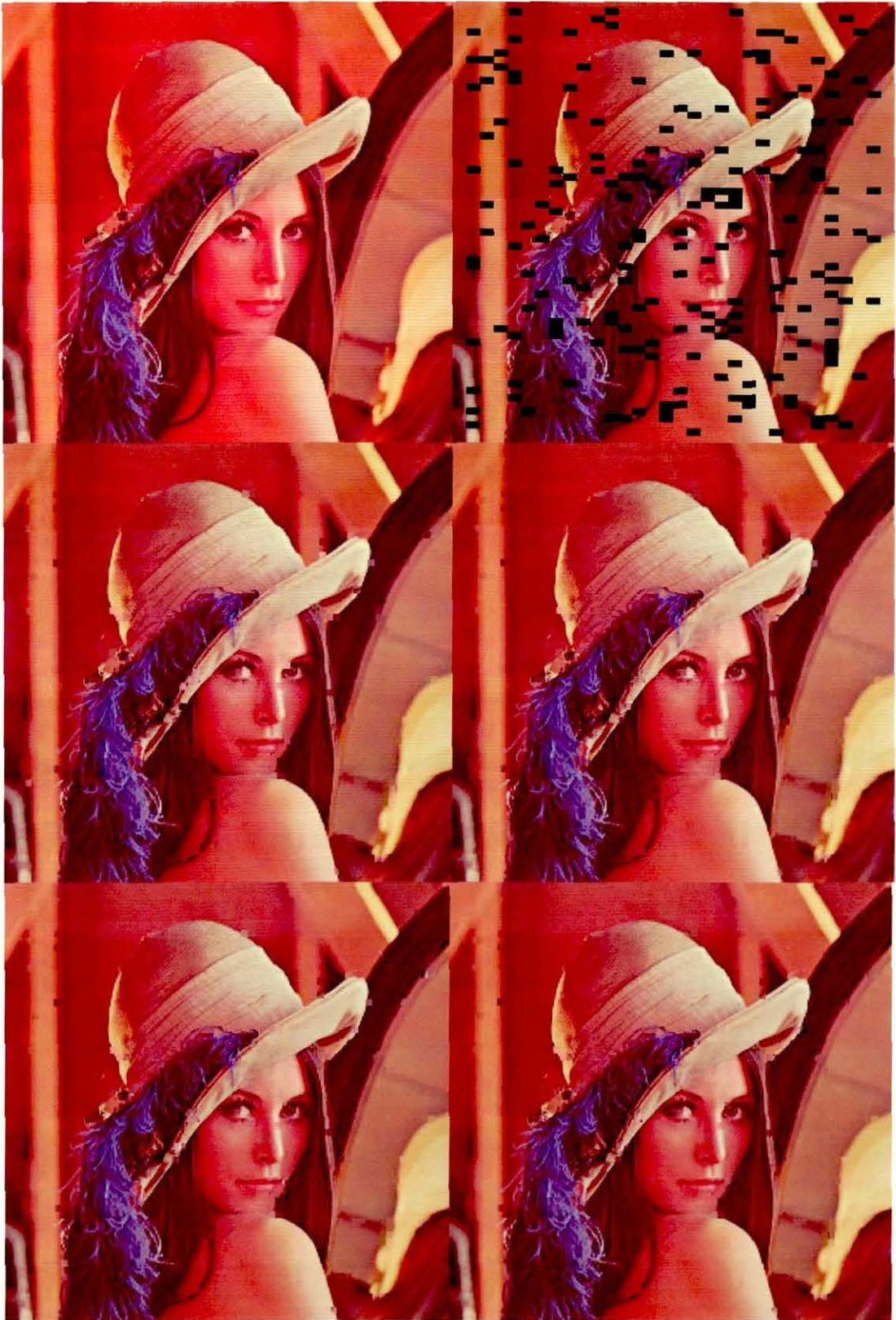


Figure 5.48

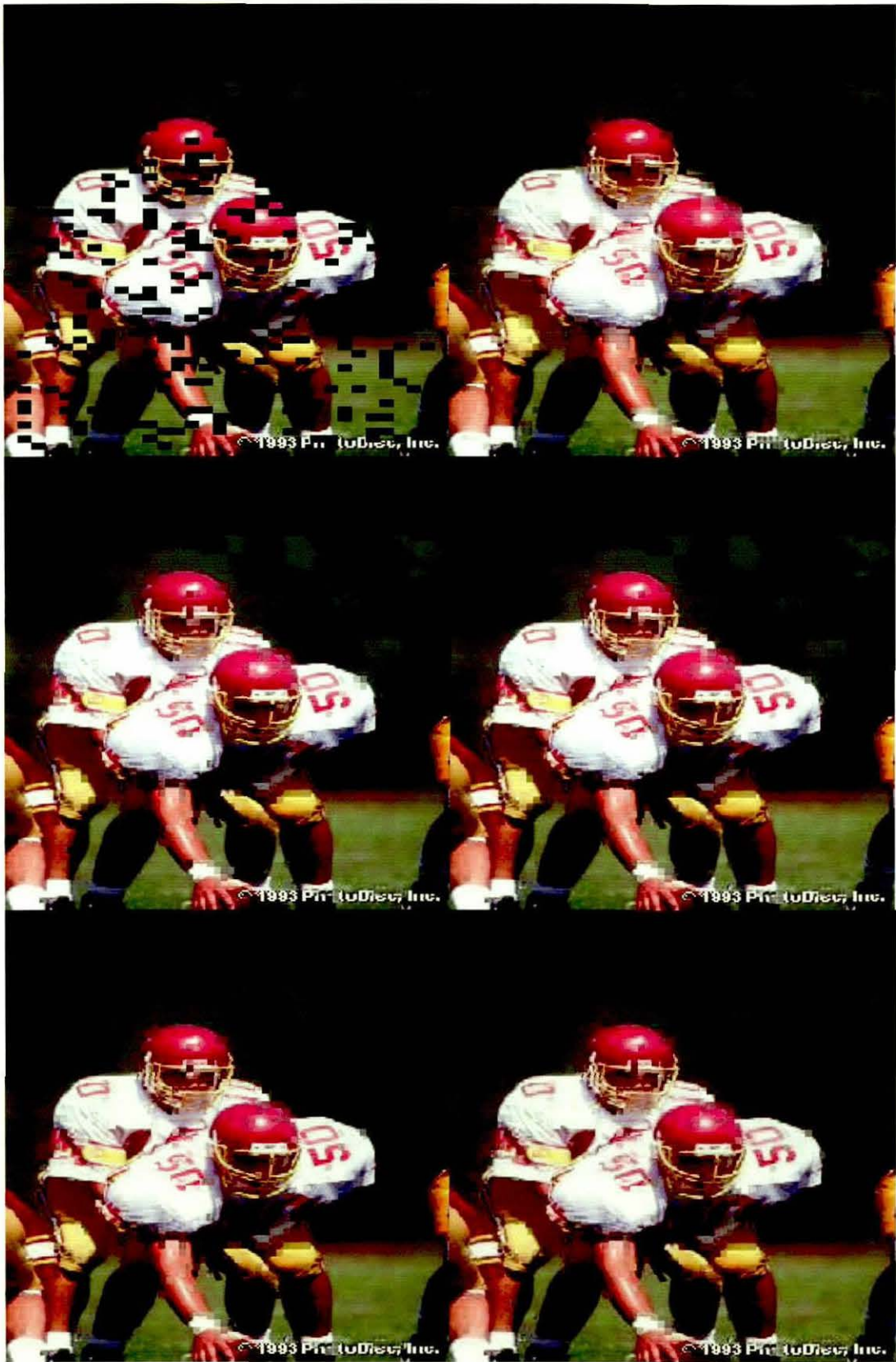


Figure 5.49



Figure 5.50

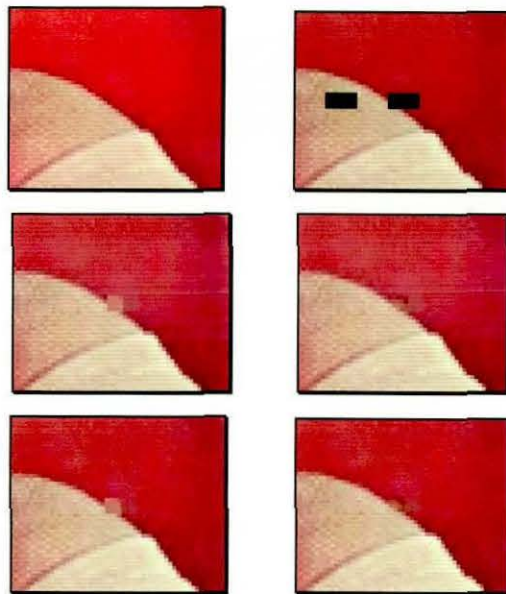


Figure 5.51

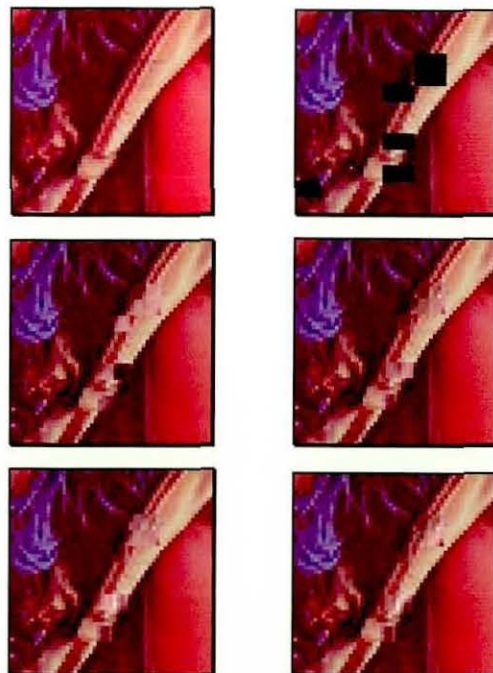


Figure 5.52

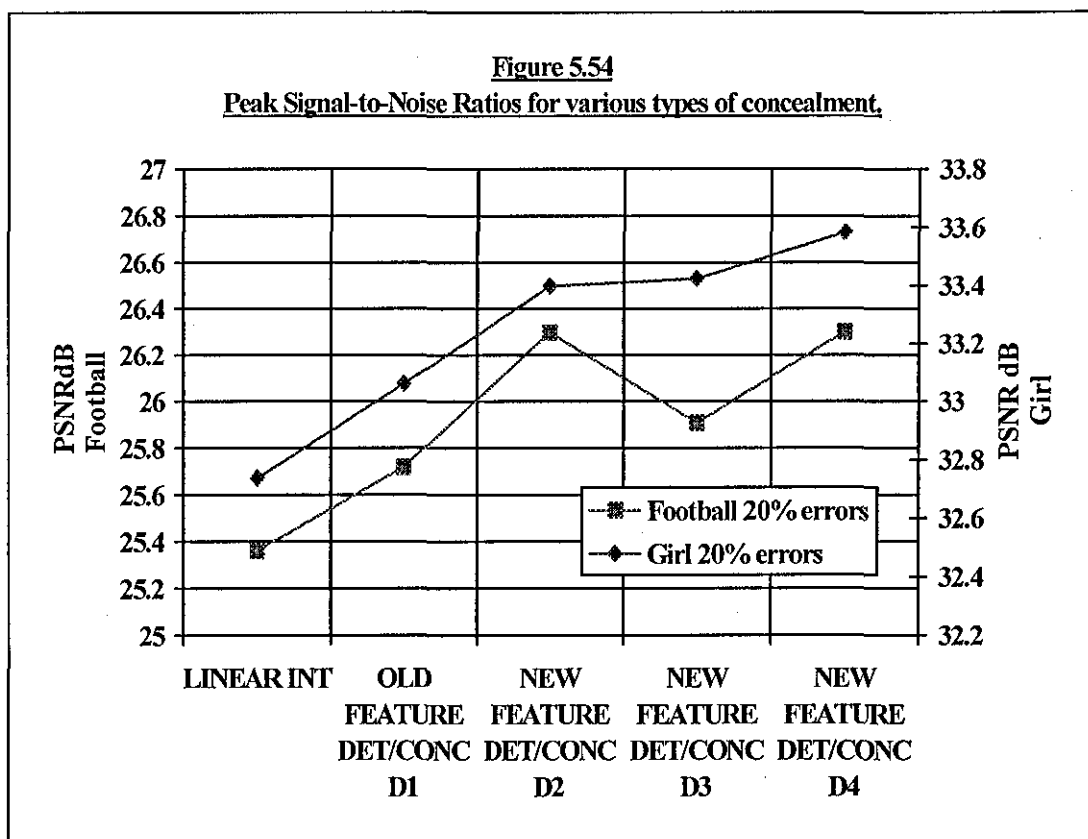
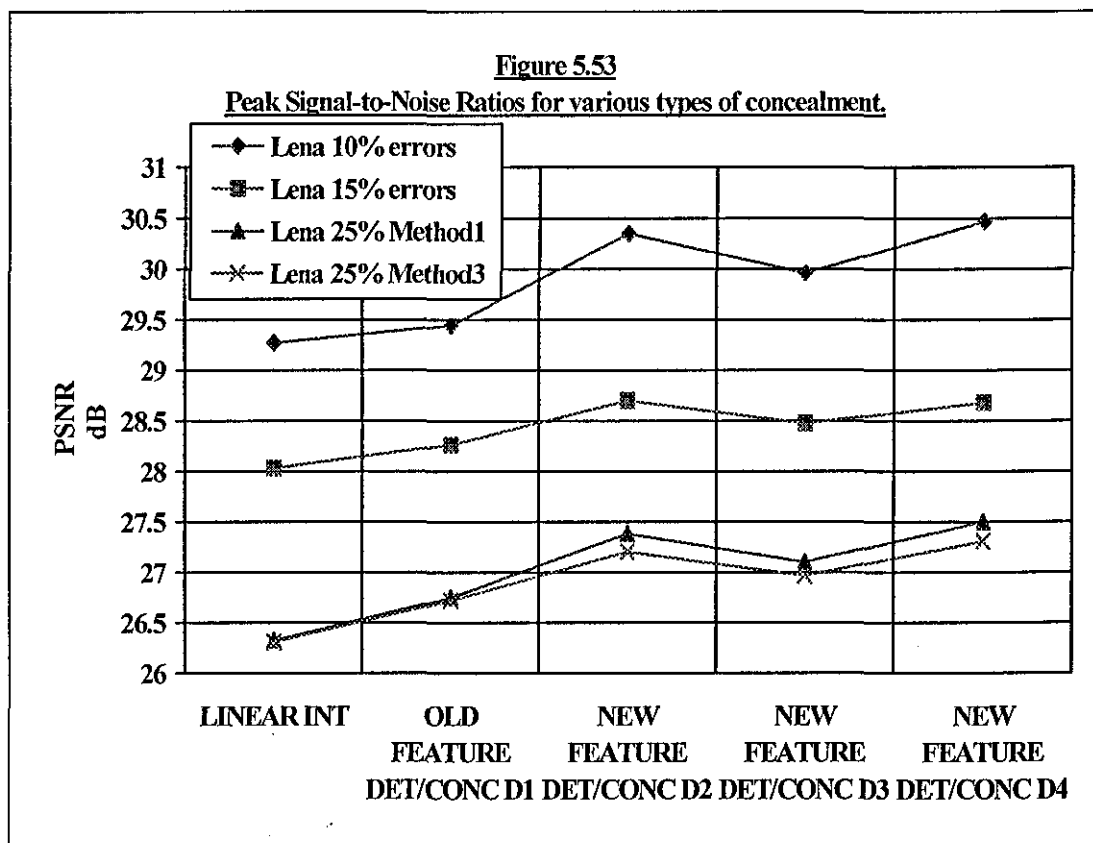


Figure 5.55
NMSE for various type of concealment in image Lena.

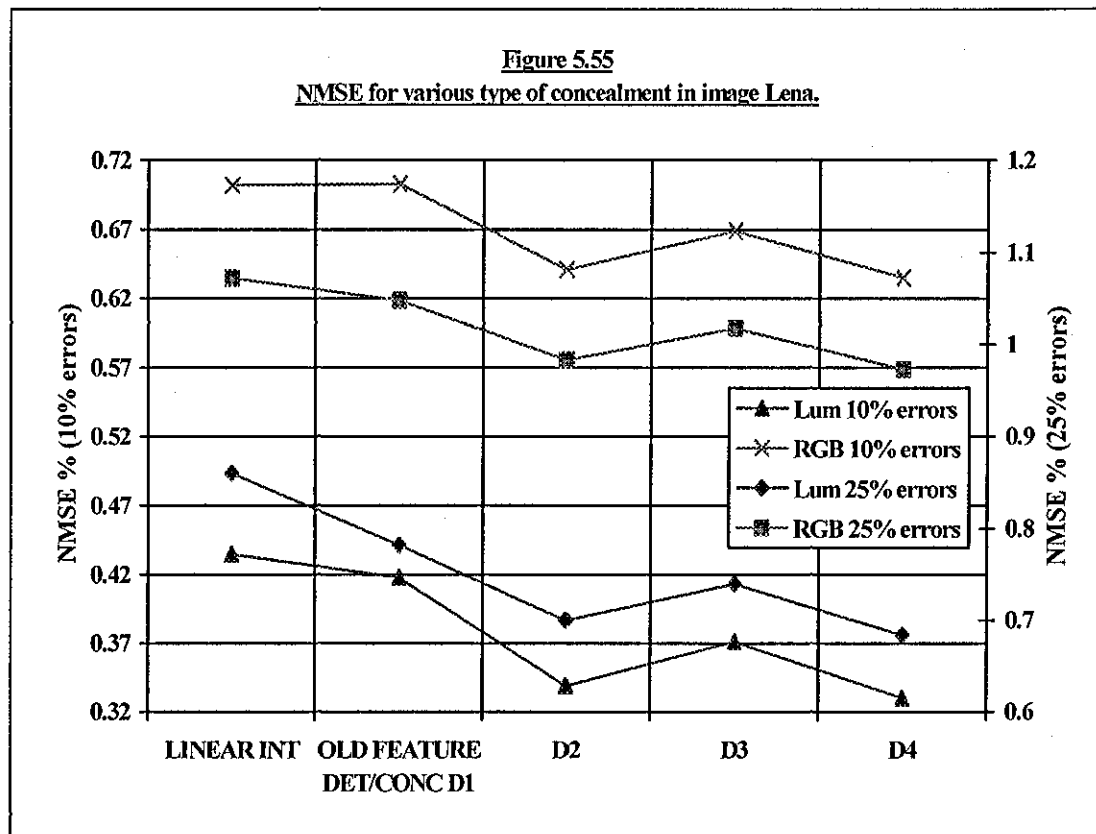
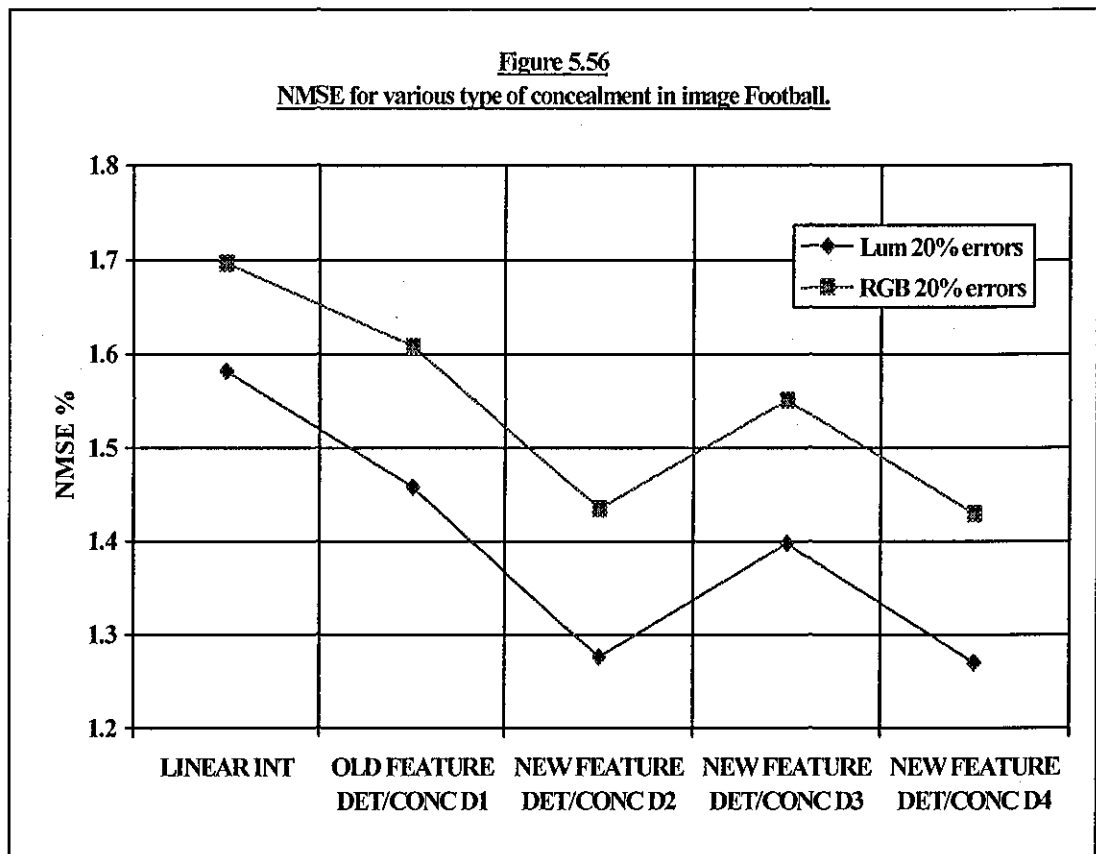
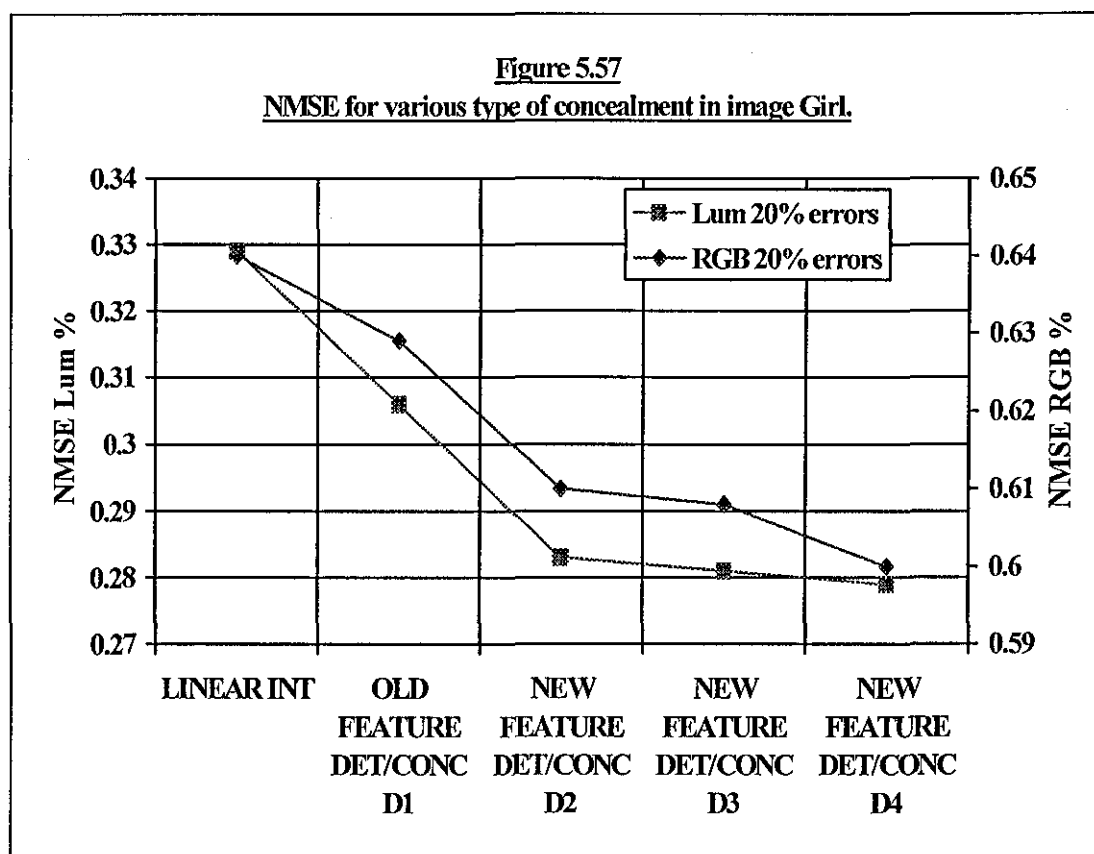


Figure 5.56
NMSE for various type of concealment in image Football.





The Images in Figure 5.47a, 5.47b all have 25% errors on average within each 512 by 512 pixel image, and images in Figure 5.49 and 5.50 all have 20% errors. The images are arranged as follows from top left to bottom right. (i) Black Block Concealment. (ii) Linear Interpolation. (iii) Initial Feature Detection/Concealment, one diagonal class D1. (iv) New Feature Detection/Concealment, with two diagonal classes D2. (v) D3, three diagonal classes. (vi) D4, four diagonal classes. The images in Figure 5.48 all have 10% errors on average and are arranged from top left to bottom right as (i) Original Image. (ii) Black Block Concealment. (iii) Initial Detection/Concealment, D1, with Intelligent Linear Interpolation. (iv) D2 classifier/concealment. (v) D3 classifier/concealment. (vi) D4 classifier/concealment. Figures 5.51 and 5.52 are similarly arranged.

The graphs in Figures 5.53 to 5.57 show the improvements possible with the new Feature Detection, Classification and Concealment algorithms. Also, the greater the number of diagonal classes the better the concealment, which is as expected as the classification errors will be smaller and this feeds down to the concealment algorithm.

However, in images Lena and Football, the D2 results are better than for D3, this is probably due to the fact that the majority of edges within these images are closer to the angles chosen for concealment in each class in D2 than the angles chosen in D3.

Figure 5.47a shows the results when $AC01(n)$, $AC10(n)$ are calculated using Method 3 and $AC11(n)$ is calculated using Method2. Figure 5.47b shows the results when $AC01(n)$, $AC10(n)$ are calculated using Method1 and $AC11(n)$ is calculated using Method1. Also, the Initial Concealment algorithm in Figure 5.47a uses 'Intelligent Linear Interpolation' as used in D2, D3 and D4 algorithms, whereas in Figure 5.47b normal Linear Interpolation is used, where all surrounding blocks are included irrespective of feature content.

In the graphs of Figure 5.55, the NMSE for RGB of the images in Figure 5.48 (Lena 10% errors), shows that the performance of the Linear Interpolation method is slightly better than that for the Initial Detection, Classification and Concealment method, whereas for the Luminance case the opposite is true. This is possibly due to the fact that as mentioned above most frequency components of a colour image are primarily in the Luminance component. Therefore, in situations where a surrounding block contains an edge that due to its displacement does not enter the error block, the Initial Detection, Classification and Concealment algorithm may place this edge into the error block. The resulting concealment is probably worse than that produced when the Linear Interpolation of all the surrounding blocks is entered into the error block. Then, when the conversion from Yuv to RGB is performed this error is distributed across the three colour image components. For all NMSE results for RGB the difference between the results for Linear Interpolation and Initial Feature Detection/Concealment, D1, is smaller than those for NMSE for Luminance are.

Figures 5.46 and 5.51 show a situation where the surrounding blocks containing the edge are not the diagonal blocks connected to the error block. Therefore, using the Initial detector, classifier and concealment algorithm, this edge is not used to aid concealment. If these blocks are used in the Initial concealment algorithm the results are very poor due to the fact that the algorithm simply interpolates the coefficients in the two surrounding blocks containing the edge. As explained earlier this does not

produce good results unless the edge is 45 degrees and the blocks used are the diagonally connected blocks. In Figure 5.51 the bottom left image shows the concealment produced using the D3 classification and concealment algorithm and as can be seen an incorrect classification has occurred, resulting in a concealment similar to that of the Initial concealment algorithm, in the middle left image. In each case there is a visual discontinuity in the edge. However, the middle right image shows the results using the D2 algorithm and as can be seen the edge has been concealed correctly. The edge has also been concealed well in the bottom right image, where the D4 algorithm was used. Figure 5.52 again shows the concealment of errors using the Initial/D1, D2, D3 and D4 concealment algorithms and as expected there is an improvement in each subsequent image.

The above examples give an indication of how each individual algorithm performs. Also, as with any classification algorithm there will be miss-classifications, but the more sophisticated the algorithm the probability of miss-classification is reduced and the amount of miss-classifications is greatly reduced when the classification is done through the use of coefficient ratios, as opposed to using the magnitude of certain transform domain coefficients.

5.3 Summary and Conclusions.

This Chapter has introduced the ideas and algorithms to perform Error Concealment within the transform domain. The calculation of four transform domain coefficients was shown, where the coefficient Ratios as described and used in Chapter 4, as well as a new ratio called Ratio4 (5.10) or (5.11), were used.

There are trade-offs when only four transform domain coefficients are calculated for the error blocks, although most are positive. The main and possibly only disadvantage is the reduced definition within the concealed edge. However, due to the fact that the human visual system is less sensitive to high frequency information and the effect of Mach Bands [36], this lack of definition is compensated for, to a certain degree. This lack of definition is also taken advantage of, to compensate for the errors introduced when the classifier groups edges into a number of pre-defined classes. This process is

analogous to the quantisation of the coefficients, with its corresponding quantisation error, which in its self removes definition from the edges within an image. This classification error has a knock on effect, where the concealment algorithm conceals for the centrally located edge within any particular class. This explains why the performance of the classification and concealment algorithm improves as the number of classes within the classifier increases i.e. the classification error is reduced.

The advantages of only calculating four coefficients are (i) Minimal amount of processing within the concealment algorithm, specifically 1 addition, 4 subtractions and 3 multiplication's. (ii) Compensation for Classification error. (iii) Compensation for Concealment error as a direct result of Classification error and Concealment algorithm error.

Interestingly, the processing load required to perform Linear Interpolation is greater than that required using the D2, D3 and D4 concealment algorithms. Therefore, the Initial detection, classification and concealment algorithm requires a greater amount of processing than that for the D2, D3 and D4 algorithms. This processing load is not proportional to the size of the pixel block, as is the case with more conventional pixel domain error concealment algorithms [6, 22, 23, 25, 26, 27, 28]. For example *Chung, Kim and Kuo* [39] proposed 'DCT Based Error Concealment for RTSP Video over a Modem Internet Connection', and state that, "One of the computationally simplest spatial Error Concealment is known as Linear Interpolation Error Concealment (LEC), LEC requires $2N^2$ multiplication's to restore an $N \times N$ block. On the other hand, the proposed DCT coefficient based spatial EC (DEC) uses $20N^2$ multiplication's to restore a similar sized loss. At first, this may seem like a substantial complexity increase, but actually it is consistent with other computations taking place side by side. For example, an IDCT for the same sized $N \times N$ block requires $12N^2$ multiplication's".

When the performance of the (DEC) algorithm is compared to the performance of the algorithm described in this Chapter, it is difficult to see any improvement. Particularly, when errors occur across diagonal edges the performance of DEC appears to be worse than for Transform Domain Feature Detection, Classification and Error

Concealment (TFDCEC), as proposed within this thesis, and definitely does not justify the additional amount of processing load required, which can be a 100 times greater for each block in error. This is due to the fact that only the macro blocks directly above, below, left and right of the error blocks are used to predict the information in the lost blocks. Therefore, if an edge is present in one of the surrounding diagonal blocks, the concealment algorithm does not use this information to aid concealment. Also, if this edge is in a block that is directly above a different error block, then it is used in the interpolation process for this error block. This again produces poor concealment by the inclusion of false edges into the error block, as seen in Figure 5.45 for the top left error block. This situation can be seen in Figure 5.26 for Edge 2 i.e. Error block L2 has an edge in the upper left diagonal block, Top L1, but uses block Top L2 in the interpolation of the coefficients in the error block. Also, Error block L1 uses block Top L1 in the interpolation process, but this block contains an edge that does not enter Error block L1, which results in the inclusion of a false edge in Error block L1. Therefore, as can be seen, using the DEC algorithm produces a visible discontinuity in a diagonal edge that crosses the error blocks and a false edge in the error block adjacent to the discontinuity, as can be seen in Figure 5.58 below.

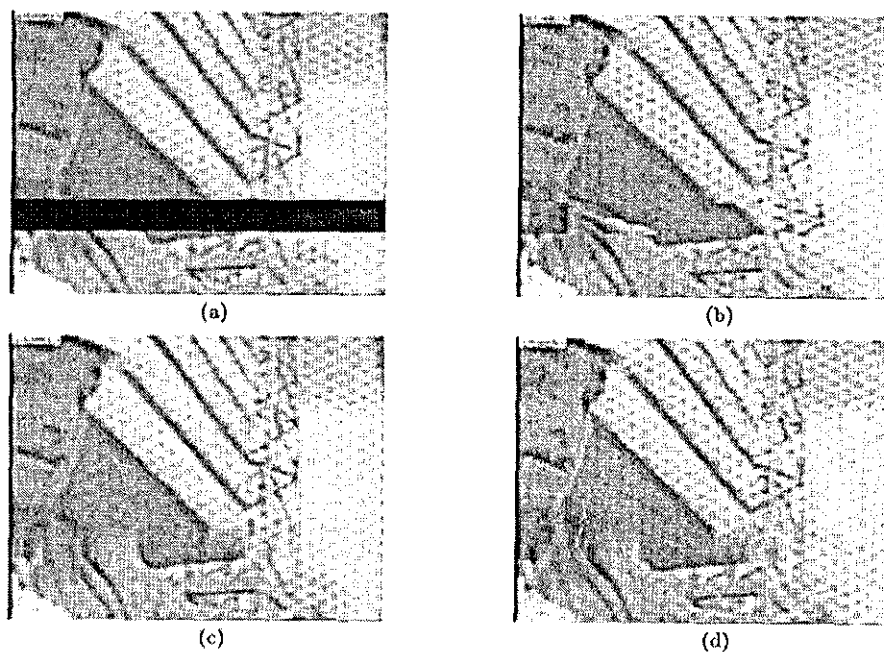


Figure 5.58

In Figure 5.58 image (a) is the original frame with the loss of a horizontal strip of blocks; image (b) is the frame where the loss has been concealed with a replication concealment algorithm, where the lost strip of blocks are replaced with those from the same position in the previous frame; image (c) is the frame where concealment is performed with a Linear interpolation algorithm (LEC), within the pixel domain, from information in the current frame; and image (d) is the frame concealed with the DEC algorithm.

CHAPTER 6

CHAPTER 6:

FURTHER DEVELOPMENTS AND CONCLUSIONS.

The research was conducted as laid out in Chapter 1. The analysis and findings were reported in detail in Chapter 2 to 5. This final chapter discusses the possibilities for future extensions and research to the work carried out within this research. This is followed by the author's conclusions on the main findings within this research, and an evaluation on whether the objectives, as laid out in Chapter 1, have been met.

6.1 Further Developments

The transform domain feature detection, classification and concealment algorithms produced within this research have shown that improvements are possible over the initial classifier of reference [31] when using advanced feature detection, classification and concealment within the transform domain for single edges. Future research could look at the effect on accuracy, complexity and the amount of processing required when the number of diagonal classes is changed from the minimum of one to a maximum possible amount using the existing algorithms. Experimenting with the size of each class within a given classifier, the inclusion of

the chrominance information into some or all of the stages of the concealment process, and the detection, classification and concealment of multi-edged features could be addressed. In addition, at present the algorithm assumes that the edge remains at the same angle through the error and surrounding blocks, however, if the edge has a different angle in the two surrounding blocks then not only does the edge within the error block have a different displacement to these blocks but the angle will also be different. Therefore, a method of calculating the edges required within the error block when this situation occurs, and how the coefficients change when the angle of the edge is changed, could be investigated. Calculation of higher frequency components could also be investigated to improve the definition of the concealed edges.

The concealment of lost blocks within P and B-frames is also a possibility as is the potential for the concealment of lost motion vectors. That is, the knowledge of how coefficients vary when an edge is displaced i.e. when there is motion in a region of an image containing an edge, may allow the motion vectors for this region to be calculated or concealed if the information is lost due to errors within the network transmission system. The knowledge of the position and angle of edges within an image can have application to many different areas, and this research has shown a mechanism to determine this information from the transform domain only. The use of the algorithms with other transforms e.g. DWHT and possibly wavelet transforms could be investigated.

Due to the fact that the concealment algorithms work on, and replace lost $N \times N$ sized blocks, there is a possibility of having blocking artefacts in the concealed areas once the frame is converted back into the pixel domain. However, there are postprocessing techniques to smooth these blocking effects only on the concealed areas [40]. This would obviously increase the processing load, but because the algorithms within this research have a minimal processing load, research into the benefits of using such techniques could be investigated.

Research into the partitioning of frames into different sized $N \times N$ blocks depending on the content of the frame e.g., areas with little edge information or texture would

use large sized blocks, and areas with edge information would use small sized blocks. The consequence of such a scheme for concealment using the algorithms developed within this research is great. That is, as the processing load is not proportional to the size of the block (as is the case with pixel domain techniques) the size of the blocks is irrelevant to the concealment stage. Whether each frame used the same size of blocks, or different sized blocks were allowed within a single frame, would also be investigated to determine if there is advantage in using either scheme, when error concealment is taken into account.

Another possible avenue of research is to investigate the 'Rate Control' or Qos filtering possibilities of forced or selected block loss before transmission at the receiver. This would require the transform domain feature detection and classification algorithm to be performed on the frame at the transmitter. Once the frame has been classified into blocks containing edges and flat blocks, a further algorithm would scan these classifications to find edges or flat blocks that could be concealed effectively with the concealment algorithms at the receiver, and therefore removed from the frame at the transmitter. This reduction in the amount of blocks being transmitted would reduce the amount of bandwidth required. The algorithm would have to ensure to remove the markers (JPEG/MJPEG) or start codes (MPEG), to ensure the receiver is capable of detecting the 'lost' blocks. There would be a small increase in the processing load at the transmitter; however, as mentioned above the algorithms produced within this research have a minimal processing load. In addition, a reduction in the transmission bandwidth reduces the probability of cell loss within the network, therefore, there is a possibility that the amount of processing required at the transmitter plus that at the receiver, would be the same or close to (or possibly less) than would be required without such a scheme. The processing load is being distributed between the receiver and the transmitter, rather than being totally at the receiver. This approach would suit the transmission of JPEG and MJPEG frames better than MPEG frames, as the encoding and decoding process is symmetrical using JPEG/MJPEG but with MPEG the encoding process is a lot more complex than the decoding process.

A further possibility due to the small processing load of the algorithms, would be to have the interconnecting nodes within the network perform transform domain feature

detection, classification and concealment on the frames. This would ensure that at the receiver the frame would contain more information than would be the case with concealment at the receiver alone. However, this scheme would require the buffering of the whole image at each node, plus the need for processing, which would not be possible using an ATM network. A similar scheme could be used when Qos filtering is performed within a large network [41].

6.2 Summary and Conclusions

There is a tendency for people to overcomplicate the solution to a given problem and introduce sophistication for it's own sake. If a problem can be solved simply, then this should be done. The performance of an image processing algorithm depends on many factors, such as the application for which the algorithm was developed and the class of images used. When real time constraints are imposed on the algorithm and possible introduction of a concealment algorithm into the decoding process, then the ratio generation technique (TFDCEC), as developed within this research has very little effect on the overall decoding time.

An improvement to the existing transform domain methods of error concealment of a video transmission system over an ATM network has been observed and introduced within this research. Each error was concealed with the transform domain feature detection, classification and concealment using the ratio generation technique.

Chapter 4 first looked at the how certain features were represented within the transform domain, by producing a number of test images. It was found that each feature e.g., horizontal edge, vertical edge and diagonal edge had a distinctive 'fingerprint' of sorts within the transform domain. This led on to the formulation of an initial transform domain feature detection and concealment algorithm, that classified the edges by inspection of the magnitudes of the first few transform domain coefficients. This algorithm produced improved concealment results when the losses fell across edges/features. However, this algorithm had a number of weaknesses i.e. only one class for diagonal edges, no calculation of displacement or position of the edge within the surrounding blocks to the lost block. These resulted in a number of miss-classifications where some edges were concealed as flat blocks and in situations

when the angle and position of a surrounding edge are such that the edge will not enter the lost block, are concealed with a falsely positioned edge. In addition, the concealment stage simply interpolated the first few coefficients of those surrounding blocks that contained the edge/feature (no displacement of the edges in the surrounding blocks). This resulted in poor concealment of all edge angles other than edges with an angle of 45 degrees. These failings helped initiate the work on the classification of blocks using the ratios of low frequency coefficients. This technique resulted in the classification of edges into a number of classes (within a range of angles) and a displacement or position value for the edge. The advantage of using ratios is that the edge contrast is not an issue in the classification of the edges, and therefore the number of miss-classification is greatly reduced.

Chapter 5 looked at the transform domain concealment of edges once the transform domain feature detection and classification algorithms have been performed. This research looked at the calculation of a new set of low frequency coefficients so as to displace the edges in the surrounding blocks, so as to produce an edge in the lost block that is in line with the surrounding edges. This improved on the initial technique described above, where the coefficients were simple interpolations of those in edges in surrounding blocks. The initial techniques i.e. linear interpolation alone and the simple feature detection and concealment algorithms often produced concealed edges with visible discontinuities due to either no edge or poor edge alignment. This was achieved by the use of the ratios used in Chapter 4, along with the analysis of the displacements required for edges in each classification and the analysis and modelling of how the coefficients vary as an edge is displaced across a given $N \times N$ block. In addition, a further ratio was used in a different method of concealment. This possibility for the use of this new ratio became apparent after the analysis of how the coefficient varied as an edge was displaced. That is, irrespective of edge contrast, the shape of the coefficient change with displacement is the same, and the ratio of one position to another is constant irrespective of contrast.

The results and objective measurements obtained in Chapter 5 show the improvements possible with the new techniques and algorithms developed within this research. However, there is still scope for improvement using such techniques, as miss-

classification although rare, still occurs and the edge alignment of some edge/feature concealment is not exact.

As has been shown in this thesis, the investigation was conducted along the research objectives set out in Chapter 1. The research objective was to produce simple/fast concealment algorithms so as to be implementable in real time, for networked video transmission systems, and a number of algorithms were produced, with a small processing load that is not proportional to the size of the blocks being used. These algorithms can be applied to any DCT based image transmission system.

APPENDIX

APPENDIX A:

THE DISCRETE COSINE TRANSFORM.

The Discrete Cosine Transform is a sub-optimum transform whose basis vectors can be shown to be closely related to the basis vectors of the optimum KLT [9]. Thus, at low orders e.g. 2×2 , the basis vectors of the DCT are identical to the basis vectors of the optimum KLT.

The DFT uses basis vectors comprising sampled sine and cosine functions, whereas the DCT uses only sampled cosine functions with special amplitude and phase relationships, which are not given by the alternating 0 and $\pi/2$ sines and cosines of the DFT. The DCT is especially significant in image coding due to its closeness to the KLT, and the fact that it is an orthonormal transform, where forward and backward transforms have identical kernels except for a scale factor. Any transform attempts to convert pixels that are statistically dependent into independent coefficients. The two-dimensional DCT is normally used in preference to the one-dimensional DCT due to

the fact that the two-dimensional transform takes account of both the horizontal and vertical correlation that exists in the image pixels. The first step in transforming an image is to divide the image into sub-blocks of a fixed size. With respect to the block size of pixels to be transformed, small sub-blocks (2×2 or 4×4) allow the system to be highly adaptive to changes in spatial detail, at the expense of reduced de-correlation performance. The reverse is true for large sub-blocks of 32×32 pixels or more. It is generally recognised that the best compromise is obtained using square sub-blocks of either 8×8 or 16×16 pixels. Once the transform coefficients have been found, compression is achieved by removing those coefficients with small energy. Therefore, the transform is required to compact most of an image's energy into as few coefficients as possible. The $N \times N$ pixels produce $N \times N$ transform coefficients, however, most of the coefficients will be zero and may be discarded. The equations for the two-dimensional DCT and inverse DCT are shown below.

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

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

Another attractive property of the DCT is its ability to approximate lines well after a number of coefficients have been discarded, unlike the FFT. An example is shown in Figure 1. The set of $N \times N$ transform coefficients are the frequency components of the image. The top left coefficient is the DC coefficient, which is a measure of the average value of the pixels in the block. All other coefficients are AC coefficients, whose frequency increases the further away they are from the DC coefficient (see Figure 2.4 in Chapter 2.

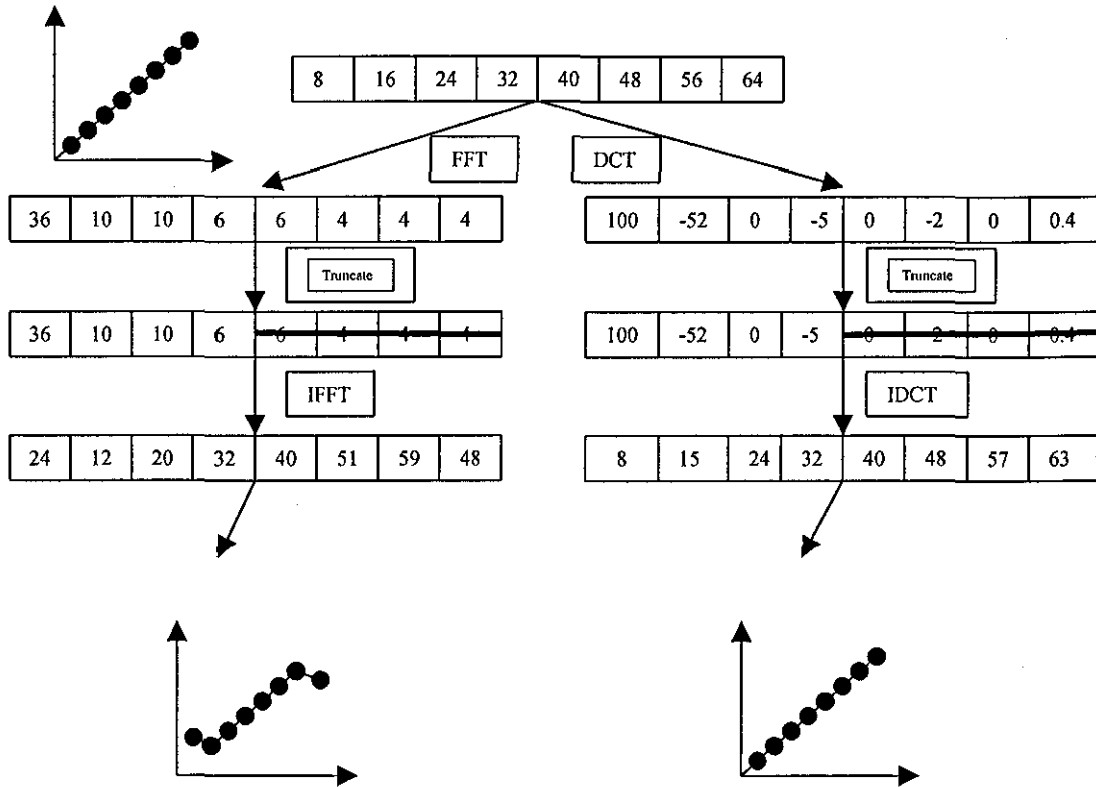


Figure 1

REFERENCES

REFERENCES:

-
- [1] M.dePrycker, "Asynchronous Transfer Mode", *Ellis Horwood Limited*, Chichester, 1992.
 - [2] U. Black, "ATM: "Foundation for broadband networks", *Prentice Hall Series in Advanced Communications Technology*, Prentice Hall, Englewood Cliffs, NJ, 1995.
 - [3] M.J.Riely, I.E.G.Richardson, "FEC and Multi-Layer Video Coding for ATM Networks", *chapter in 'Performance Modelling and Evaluation of ATM Networks' ed. D.Kouvatsos*, pub. Chapman & Hall 1995, pp 450-457.
 - [4] M.J.Riely, B.U.Kuhler, I.E.G.Richardson, "A Variable Block Length FEC Protocol for MPEG Packet Video", *Proc. 3rd International Symposium on Communication Theory and Applications*, Lake District, July 1995.

- [5] D.R.Bull, J.T.Chung, "Error-resilient image and video coding for wireless communication systems", *Electronics & Communication Engineering Journal*, pp. 181-190, August 1998.
- [6] H.Sun, J.Zdepski, "Adaptive Error Concealment Algorithm for MPEG Compressed Video", SPIE, *Visual Communications and Image Processing*, vol. 1818, pp. 814 - 824, 1992.
- [7] M.Ghanbari, "Two-layer coding of video signals for VBR networks." *IEEE J.Select. Areas Commun*, vol. 7, pp.771-781, June 1989.
- [8] R.J.Clark, "Transform Coding of Images", Academic Press, London, 1985.
- [9] N.S.Jayant, P.Noll, "Digital Coding of Waveforms", Prentice Hall, Englewood Cliffs, NJ, 1984.
- [10] K.R.Rao, "Discrete Transforms and their Applications", Van Nostrand Reinhold, New York, 1985.
- [11] 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.
- [12] M.Covell, "Low data-rate video conferencing", Master's Thesis, Dept. of EECS, MIT, Dec 1985.
- [13] R.J.Clark, L.M.Linnet, "Fractal and image representation", *IEE Electronics and Communication Engineering Journal*, vol. 5, No. 4, Aug 1993.
- [14] D.A.Huffman, "A method for the construction of minimum redundancy codes", *Proc. IRE*, vol. 40, pp. 1098-1101, Sept 1952.

- [15] G.K.Wallace - "Overview of the JPEG (ISO/CCITT) Still Image Compression Standard", SPIE, *Image Processing Algorithms and Techniques*, vol. 1244, pp. 220-233, 1990.
- [16] (JPEG) ISO DIS 10918-1, "Digital Compression and Coding of Continuous-tone Still Images", ISO/IEC JTC1/SC2/WG10, Jan 1992.
- [17] "MPEG-2 Editing Solutions Unveiled", TV Technology, April 20, 1998.
- [18] *CCIT Recommendation H.261*, "Video Codec for Audio-Visual Services at p*64 kbit/s", Dec 1990.
- [19] (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.
- [20] G.C.Kessler, P.Southwick, "ISDN Concepts, Facilities and Services", *McGraw-Hill Series on Computer Communications*, McGraw-Hill, third ed., 1997.
- [21] M.Bystrom, V.Parthasarathy, J.W.Modestino, "Hybrid Error Concealment Schemes for Broadcast Video Transmission Over ATM Networks", *Electrical, Computer and Systems Engineering Dept.*, Center for Image Processing Research, Rensselaer Polytechnic Institute, Troy, New York.
- [22] S.Aign, K.Fazel, "Error detection and concealment measures in MPEG-2 video decoder", *Proceedings of the International Workshop on HDTV*, Torino, Italy, October 1994.
- [23] J.F.Shen and H.M.Hang, "Compressed image concealment and postprocessing for digital video recording", *Proceedings of the IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 636-641, Taipei, Taiwan, December 5-8 1994.

- [24] M.Ghanbari, V.Seferidis - "Cell-loss concealment in ATM video codecs", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 3, Iss. 3, pp. 238-247, June 1993.
- [25] W.Kwok, H.Sun, "Multidirectional interpolation for spatial error concealment", *IEEE Transactions on Consumer Electronics*, vol. 3, No. 39, pp. 455-460, August 1993.
- [26] H.Sun, W.Kwok "Concealment of damaged block transform coded images using projections onto convex sets", *IEEE Transactions on Image Processing*, vol. 4, No. 4, pp. 470-477, April 1995.
- [27] D.C.Youla, H.Webb, "Image restoration by the method of convex projections: Part 1-theory", *IEEE Transactions on Medical Imaging*, vol. MI-1, No. 2, pp. 81-94, October 1982.
- [28] W.Zeng, B.Liu, "Geometric-structure-based directional filtering for error concealment in image/video transmission", *SPIE, Wireless Data Transmission, Photonics East'95*, vol. 2601, October 1995.
- [29] 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", *5th Int. Workshop on Packet Video*, Berlin, Germany, 1993.
- [30] L.T.Chia, D.J.Parish, J.W.R.Griffiths - "On the Treatment of Video Cell Loss in the Transmission of JPEG Images", *Computers and Graphics*, vol. 18, No. 1, 1994.
- [31] 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 1994*.

- [32] W.Luo, M.ElZarki, "Analysis of error concealment schemes for MPEG-2 video transmission over ATM based networks", *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, vol. 2501/3, pp. 1358-1368, Taipei, Taiwan, May 1995.
- [33] A.S.Tom, C.L.Yeh, F.Chu, "Packet video for cell loss protection using deinterleaving and scrambling", *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 2857-2860, Toronto, Canada, May 1991.
- [34] Q.Zhu, Y.Wang, L.Shaw, "Coding and cell loss recovery in DCT based packet video", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, No 3, pp. 248-258, June 1993.
- [35] A.K.Jain, "Fundamentals of Digital Image Processing", Prentice Hall, Englewood Cliffs, NJ, 1989.
- [36] J.S.Lim, "Two-Dimensional Signal and Image Processing", page 509, Prentice Hall, Englewood Cliffs, NJ, 1989.
- [37] W.-M.Lam, A.R.Reibman - "An Error Concealment Algorithm for Images Subject to Channel Errors.", *IEEE Trans. Image Processing*, vol.4, no.5, pp. 533-542, 1995.
- [38] Y.Wang, Q.-F.Zhu - "Signal Loss Recovery in DCT-based Image and Video Codecs.", SPIE, *Visual Communications and Image Processing*, vol. 1605, pp. 667-678, 1991.
- [39] J.Park, J.Kim, S.Lee, "DCT Coefficient Recovery Based Error Concealment Technique and Application to the MPEG-2 Bit Stream Error", *IEEE Cir & Sys for Video Technology*, Vol. 7, pp. 845-854, December 1997.

- [40] B.Ramamurthi, Al.Gersho, "Nonlinear Spacevariant Postprocessing of Block Coded Images", *IEEE Transactions on Acustics, Speech and Signal Processing*, vol. ASSP-34, No 5, pp. 1258-1267, October 1986.
- [41] N.Yeadon, "Qos Filters: Supporting Heterogeneous Internetworking", *Multi-Service Networks '96*, Cosener's House, Abingdon, 1996.



