High Ratio Wavelet Video Compression through Real-Time Rate-Distortion Estimation

by

Edmund Stephen Jackson
BScEng (Electronic) *Summa Cum Laude*
Student Member IEEE

Submitted in fulfilment of the requirements for the
Degree of Master of Science in Electronic Engineering
in the School of Electrical, Electronic and Computer
Engineering at the University of Natal, Durban

July 2003

Abstract

The success of the wavelet transform in the compression of still images has prompted an expanding effort to exercise this transform in the compression of video. Most existing video compression methods incorporate techniques from still image compression, such techniques being abundant, well defined and successful. This dissertation commences with a thorough review and comparison of wavelet still image compression techniques. Thereafter an examination of wavelet video compression techniques is presented. Currently, the most effective video compression system is the DCT based framework, thus a comparison between these and the wavelet techniques is also given.

Based on this review, this dissertation then presents a new, low-complexity, wavelet video compression scheme. Noting from a complexity study that the generation of temporally decorrelated, residual frames represents a significant computational burden, this scheme uses the simplest such technique; difference frames. In the case of local motion, these difference frames exhibit strong spatial clustering of significant coefficients. A simple spatial syntax is created by splitting the difference frame into tiles. Advantage of the spatial clustering may then be taken by adaptive bit allocation between the tiles. This is the central idea of the method.

In order to minimize the total distortion of the frame, the scheme uses the new ρ-domain rate-distortion estimation scheme with global numerical optimization to predict the optimal distribution of bits between tiles. Thereafter each tile is independently wavelet transformed and compressed using the SPIHT technique.

Throughout the design process computational efficiency was the design imperative, thus leading to a real-time, software only, video compression scheme. The scheme is finally compared to both the current video compression standards and the leading wavelet schemes from the literature in terms of computational complexity visual quality. It is found that for local motion scenes the proposed algorithm executes approximately an order of magnitude faster than these methods, and presents output of similar quality. This algorithm is found to be suitable for implementation in mobile and embedded devices due to its moderate memory and computational requirements.

Preface

The research described in this dissertation was performed at the University of Natal, Durban over the period February 2002 until July 2003 by Mr. Edmund Jackson under the supervision of Professor Roger Peplow.

This work has been generously sponsored by Thales Advanced Engineering and Armscor.

The financial assistance of the Department of Labour (DoL) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the DoL.

I hereby declare that all the material incorporated in this thesis is my own original unaided work except where acknowledgment is made by name or in the form of a reference. The work contained herein has not been submitted in whole or part for a degree at any other university.

Signed:

Name:

Date:

As the candidate's supervisor I have approved this thesis for submission.

Signed:

Name:

Date:

Acknowledgements

I would like thank my supervisor, Professor Roger Peplow, for allowing me the freedom to follow my research interests, as well as his encouragement and very insightful advice. I would also like to thank Professor P.G.L. Leach of the School of Pure and Applied Mathematics for helping me informally with the mathematics involved in this project.

Furthermore, I would like to thank Mr Peter Handley who has been extremely generous in supporting this work strategically, financially and technically. His patience and trust are greatly appreciated.

I also wish to thank my mother, Siobhan, my sister, Nuala, and my brother, Pearce, for their constant support and encouragement. Their tolerance of my obsessive behaviour is saintly. Special thanks are due to my mother who proof read this thesis. I would also like to thank my late father, Gerard for his example of perseverance.

I would like to thank Lao Tze Bob Davies and Shaz Davis for providing a hard rock in the rising sea.

Finally, I would like to thank all my fellow post-graduate students and members of staff in the department for their companionship. Their friendship has made these months greatly enjoyable.

Domo Arigato Gozaimashita.

Publications

The following publications are based on the work reported in this thesis.

E. Jackson, R. Peplow, *"Fast Rate-Distortion Estimation and Optimization for Wavelet Video Coding,"* 3rd International Symposium on Image and Signal Processing and Analysis - IEEE, Rome, Italy, 2003.

E. Jackson, R. Peplow, *"Video Compression System for Mobile Devices,"* South African Telecommunications, Networking and Applications Conference (SATNAC), George, South Africa, 2003.

Table of Contents

Table of Figures

Table of Tables

Table of Acronyms

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
bpp	Bits Per Pixel
CD	Compact Disk
CIF	Common Intermediate Format
dB	Decibel
DCT	Discrete Cosine Transform
DSP	Digital Signal Processor
DVD	Digital Versatile Disk
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding and Optimal Quantisation
ECECOW	Embedded Conditional Entropy Coding of Wavelets
EQ	Estimation Quantisation
ESCOT	Embedded Set Coding and Optimal Quantisation
EZW	Embedded Zerotree Wavelet
fps	Frames per Seond
GGD	Generalised Gaussian Distribution
GHz	Gigahertz
GOF	Group of Frames
GPRS	Generalised Packet Radio Service
HDTV	High Definition Television
IEEE	Institure of Electrical and Electronic Engineers
IFT	Iterated Function Transform
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
ITU	Interational Telephonic Union
JPEG	Joint Photographic Experts Group
kbps	Kilobits per Pixel
KLT	Karhunen Louve Transform
LAN	Local Area Network
LIP	List of Insignificant Sets
LIS	List of Insignificant Pixels
LSP	List of Significant Pixel
ME/MC	Motion Estimation / Motion Compensation
MPEG	Motion Pictures Expert Group
ms	Millisecond
MV	Motion Vector
OBME/MC	Overlapping Block ME/MC
PACC	Partitioning, Aggregation and Conditional Coding
PC	Personal Computer
PDA	Personal Digital Assistant
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
RAM	Random Access Memory
RD	Rate Distortion

SFQ	Space Frequency Quantisation
SNR	Signal to Noise Ratio
SPIHT	Set Partitioning in Hierarchical Trees
SRC	Stack Run Coding
STFT	Short Time Fourier Transform
TCQ	Trellis Coded Quantisation
TCSFQ	Trellis Coded Space Frequency Quantisation
TV	Television

Chapter 1 - Introduction

The complexity of information communicated by electronic means is constantly increasing. The internet was originally a text only medium and has subsequently been used to transmit still images, then audio, followed by pre-recorded video, and recently real-time video. Mobile networks demonstrate a similar trend, with the latest generation of cellular handsets capable of capturing and transmitting still images. It is projected that in the near future these devices will communicate video data.

The volume of data comprising a video stream is enormous. A simple 8 bit per pixel monochrome video sequence at the standard QCIF resolution of 176x144 pixels demands 2 Mb/s at 10 fps. This is beyond the capability of current fixed line or mobile networks to provide reliably. Thus, compression is a prerequisite for video communication. Currently, video compression technology is capable of producing a 'reasonable' quality QCIF 10 fps stream, at 20 kbps.

2.5G and 3G mobile cellular networks continue to increase the available channel bandwidth with the aim of supporting the deployment of multimedia services. Standard 2G GSM networks allow data communication at up to 9.6 kbps. Video cannot be transmitted at reasonable resolution and frame rate over such a bandwidth-limited channel using existing compression schemes. GPRS services have increased the available bandwidth to 115 kbps, which is sufficient for transmission of compressed video, at a resolution suitable for display on mobile devices, such as QCIF. 3G networks will support wide area data rates of 384 kbps. In addition wireless LAN protocols such as Bluetooth and IEEE 802.11b provide ample bandwidth to devices such as PDA's for the transmission of video using existing compression methods.

However, despite the bandwidth sufficiency of modern cellular and wireless LAN channels, devices supporting video transmission are not extant. This is mainly due to the extreme computational burden of video compression. The standard video compression systems such as H.263, H.263+, MPEG-2 and MPEG-4 all rely on complex motion estimation schemes to achieve compression. The computing power required to execute these algorithms is evidenced by personal computers being able to compress video in real-time only in recent times. The computational resources available to typical mobile devices are significantly less abundant than to a PC. Although custom video compression chipsets are available, these have not been incorporated in general technology due to cost, power and size constraints. This technical barrier has prevented the deployment of video-capable devices in cellular, and other mobile networks.

The work presented in this thesis aims to overcome this limitation by abandoning the present motion estimation and compensation paradigm, in order to produce a video compression algorithm capable of execution on a standard DSP.

1.1 Video and Wavelets

This section will briefly define video and discuss how wavelets have advanced from a mathematical curiosity to an engineering tool in the field of image and video compression.

1.1.1 Images and Video

A still image on a computer consists of a rectangular matrix of points, called pixels. Each pixel has associated with it a number that represents the colour of the corresponding point in space on the image. We will only consider greyscale images here, and in this case the pixel number represents the luminance, or intensity, of the image point. A video sequence is a series of frames, each of which is a still image. Each frame is the spatial representation of the scene at a particular point in time. If the frames are sufficiently close together, the human brain interpolates the still images into a moving scene. This arrangement is known as the raw video representation. This is the representation required to communicate video information between a machine and a human.

Any relationships within a frame are called intra-frame relationships, and are usually concerned with spatial or space-frequency concepts. Relationships between frames are called inter-frame relationships and are usually related to temporal or time-frequency notions. The relationships of interest for this work are similarities, or stated differently, correlation structures. The art and science of compression is to describe an image or video succinctly, in terms of these correlation structures. This compressed representation is used to store and transmit images and video between machines. It is evidence of the ascendancy of the human mind that the bandwidth of raw video, through which we perceive, is truly massive by comparison to the compressed representation required to allow machines to communicate timeously.

The ongoing effort in the image and video processing research community is to expose these correlation structures on one hand, and provide syntax for expressing them on the other. This vague brief has led to a plethora of techniques, standards and methods being developed.

The first class of compression schemes is termed lossless coding, as the original image (or frame sequence) may be recovered exactly from the compressed representation. The simplest such scheme is run length coding. Here, a sequence of repeated pixels is simply replaced by the pixel value and the number of times is repeated. The next level of complexity is provided by entropy based methods. Here a codebook is used to match pixel values with binary codes. The codes have variable length, and the codebook is designed so that the shorter codes are used to represent frequently occurring pixel values. This is the concept behind Huffman codes, which minimise the final stream length, under this scheme. A similar idea is used by arithmetic codes, but this is not explored here for space reasons.

Other compression schemes introduce distortion through the process of compression, and are known as lossy. Naturally such schemes are capable of providing greater compression ratios than lossless schemes. The simplest lossy compression is scalar quantisation; which simply amounts to reducing the precision of the coefficient representation. Additional considerations with scalar quantisation include variable quantisation levels and progressive quantisation. Where scalar quantisation schemes provide a code for a single pixel, vector quantisers consider sequences of coefficients. In general, vector quantisers replace a sequence of pixels (a vector) with a similar vector chosen from a codebook. The codebook is designed to minimise the total squared difference (termed distance) between the original vectors and those in the codebook. This is an extremely complex, and usually an adaptive process.

A multitude of other ideas may be found in the literature, such as [Effe98] and [Sola97] and many other books.

Often the spatial domain does not expose all the correlations that may exist. Thus several transforms have been explored such as the Fourier transform, the closely related discrete cosine transform (DCT), the Karhunen-Louve transform (KLT), the discrete wavelet transform (DWT), the iterated function transform (IFT, commonly known as fractals), and many others.

The total signal energy is the sum of the square of the magnitude of each pixel (or coefficient) in the image. In the transform domain, fewer coefficients may posses a greater proportion of this total. This is termed energy concentration. This aids compression through quantization as low energy coefficients may be coarsely quantised or discarded in the transform domain and upon the inverse transformation to the original spatial domain the fidelity decrease may be minimal. Roughly speaking, as the signal energy is concentrated, it is easier to describe.

The image compression process that is currently entrenched is based on the DCT. Details on the DCT may be found in the literature, as before. However, the DCT may be thought of as operating similarly to the Fourier transform, in that it extracts frequency information. The standard JPEG image compression system segments an image into 8x8 pixel blocks. The DCT is then applied to each block individually. It is found that in images the signal energy is concentrated in the low frequencies (which represent the average colour of a block), thus lossy compression is achieved by quantising and removing coefficients, in decreasing order of frequency. There is much more to this process, but it may be seen that the DCT exposes the short term spatial correlation between pixels. Syntax for describing the correlation is created through the blocking, and subsequent actions of raster scanning and entropy coding, which are not explored here.

MPEG is the standard video compression method, and is closely related to JPEG. The intra-frame coding is handled very similarly to JPEG, but the inter-frame aspect is unique to video coding. The method used is called motion estimation and compensation (ME/MC). Subsequent frames are compared, in terms of the blocks (usually 16x16 in this case). A block matching search is undertaken for each block in the temporally prior (original) frame. For each block, the block in the temporally latter frame which most closely matches that from the prior frame is termed the destination block. The block displacement between the original and destination block is termed the motion vector. This process is termed motion estimation, as it indicates roughly how objects in the sequence have moved between frames. A frame is constructed where each block is the block from the original frame, displaced by the calculated motion vector, this is called the motion compensated frame. The difference between this compensated frame, and the temporally latter frame from the sequence is calculated, and termed the residual frame. The signal energy in the residual frame is typically very small, thus aiding compression as there is less information. This process is explored more in Chapter 3.

1.1.2 Wavelets

This status quo has recently been under assault by wavelet transform based methods. The wavelet transform is considered a recent development in wavelets, with research into the field having started in the early 1980s [Daub92]. An introduction to the wavelet transform is given in Appendix A, with appropriate references, a brief conceptual outline is considered below.

Signals captured from the real world are usually parameterised in time or in space. For instance an audio signal is typically expressed as a function of time, an image as a function of space (x and y), and a video as a function of both space and time. Transforms change the parameterisation; the Fourier transform provides frequency information, but all spatial or temporal information is unavailable in this domain. Thus the original signal domain provides exact spatial or temporal information, but no spectral information and the Fourier transform provides the exact opposite case. The wavelet transform behaves differently by providing approximate space/time and spectral information. Thus knowledge of a signal's exact behaviour in only one domain is traded for approximate information of the signal in two domains. This has proven extremely powerful for many applications. Consider an image briefly; an edge in an image may be considered as a step function, which is easily represented in the spatial domain, but requires an infinite number of coefficients in the spectral domain. Equally, an object's interior is well represented by a several low frequency coefficients in the spectral domain, but requires many coefficients to describe in the spatial domain. The wavelet transform provides the opportunity to describe both these image behaviours in a succinct fashion.

The wavelet transform was initially considered as a continuous function, much like the Fourier transform is continuous. Interest for engineers was spurred in 1989 when Mallat developed a discrete wavelet transform using FIR filter banks, based on multiresolution analysis [Mall89]. This may be considered to be analogous to the discrete Fourier transform. Interest rapidly soared, and the application to image coding was quickly recognised [Mall89,2]. The seminal paper for this period, [Anto92], describes an iterated FIR filter bank which produces a subband decomposition of an image. Thereafter each subband is coded using vector quantisation. For approximately the next decade image compression using wavelets followed this model of subband coding. The next breakthrough came in 1992 with Shapiro's publication of [Shap93]. This paper describes a hitherto unrecognised correlation structure in images called zerotrees, a statistical inter-subband behaviour of low value coefficients. The algorithm he describes, EZW, outperforms JPEG substantially. Following this paper, Said and Pearlman published [Said96]; the famous SPIHT algorithm. This algorithm provides a better syntax for describing Shapiro's zerotree structure. Work based on zerotree coding using rate-distortion theory, joint application with other quantisation strategies as well as many other ideas continues to this time. Many such techniques are described in Chapter 3. The power of wavelets for image compression has been recognised by the JPEG body, with the recent standardisation of JPEG2000 (Section 3.3.12), which is based on wavelets.

Following the general trend, once wavelets had proven useful in image processing their utility for video compression was brought into consideration. The natural approach was to use the ME/MC approach from MPEG-like coding to convert wavelet image coders into video coders. However, several problems with this approach have been identified. Generally, wavelet image coders consider the entire image, rather than segment the image into blocks which is common to

DCT type coders. For still image coding this has meant that the infamous blocking associated with JPEG has been avoided, leading to better still image compression. However, for video compression the blocks have provided a useful spatial syntax for ME/MC, as explored above. Some algorithms have reintroduced blocking in order to gain advantage through block ME/MC. Attempts to code the blocks individually using wavelet techniques have proven unsuccessful, as the dataset is too small to allow the zerotree structure to express fully. Coding entire block ME/MC compensated frames has also proven troublesome as the edge effects between blocks interfere with the quantisation assumptions of the coders. Nevertheless the use of edge reducing filters, as well as other techniques, has provided gains. Another approach is to include the temporal dimension into the wavelet transform, thus creating a 3D transform. This approach is discussed along with others in Chapter 4. Although wavelet video compression algorithms of similar or mildly superior performance to existing standards have been developed, the massive advantage held by wavelets in image compression has yet to translate to video coding. The 'grand challenge' is to find a means of describing motion in a wavelet-friendly manner.

1.1.3 The Rate Distortion Theory

An underlying theme in most source coding is the rate-distortion (RD) theory. This theory states that there exists a relationship between the number of bits used to describe a source (the rate) and the fidelity of the corresponding representation (the introduced distortion). The main goal of this research field is to predict the RD behaviour of a source in a fast and accurate fashion. In general the methods proposed to date have been able to fulfil one of these requirements at the expense of the other.

1.2 Roadmap of Thesis

The first four chapters of this thesis present the literature review that was undertaken in preparation for designing the algorithm. The following four chapters build on specific observations drawn from this review to propose a new algorithm, describe it, and quantify its performance

Chapter 2 presents a brief overview of the rate distortion theory. The goal of the chapter is to introduce a recently published algorithm that performs accurate and fast RD estimation; this algorithm is crucial to the algorithm proposed in subsequent chapters. The background theory and other algorithms are presented in order to provide context to this algorithm. In addition, this chapter presents RD optimisation; techniques of using the RD estimate of a source to compress it optimally.

Chapter 3 presents a broad literature review of wavelet image compression techniques. Although this work produces a video compression technique, it was found that all the wavelet video compression techniques presented in the literature rely on still image counterparts for spatial decorrelation. Thus, in order to understand these video techniques, it was necessary to review still image compression techniques, and this review is presented here.

Chapter 4 presents video compression techniques. The current state of the art is presented through a brief review of the H.263 and MPEG standards. Thereafter a review of wavelet video compression schemes proposed in the literature is presented, and compared to the standards. This chapter highlights the challenges faced by the wavelet transform in competing with the DCT techniques, owing to the difficulty of motion estimation using the wavelet transform.

Appendix A presents a mathematical review of the wavelet transform. Many of the papers encountered during the literature review rely on concepts that are specific to the wavelet transform. Thus it was necessary to review this theory in order to understand its application. Appendix A does not form part of the thesis argument, but is presented for completeness and reference.

These chapters form the theoretical background to the project. RD theory is a thread running through all source coding, and so it presented first, in Chapter 2. Image coding is a pre requisite to video coding and thus is presented thereafter in Chapter 3. Finally video coding, which is the realm of this project, is presented in Chapter 4. Although this project never deals explicitly with wavelets Appendix A is provided introducing the theory. This is often necessary for reference in Chapter 3, where the properties of the wavelet representation are exploited. Hence the first three chapters and Appendix A provide a cogent development of the literature.

Having undertaken this exhaustive literature review, the new technique was designed. Chapter 5 presents the premise for the algorithm. In this chapter the computational load of motion estimation and compensation is quantified. In addition, our proposed alternative of difference frame coding is explored. The properties of difference frames are investigated, and means of their exploitation are given.

Chapter 6 presents the proposed algorithm in its entirety. The initial focus of the chapter is the system level design of the algorithm to meet the specified criteria. The remainder of the chapter is dedicated to implementation details, specifically where departures from the literature are made.

Chapter 7 presents the performance of the new algorithm. Both the computational load and quality of the output stream are compared to the H.263 and MPEG standards, as well as the other wavelet techniques presented in the literature. In addition the various assumptions made during the algorithmic design are tested.

While chapters 2 through 4 provide a theoretical backdrop and context to the project, chapters 5 through 7 develop and test the algorithm in question. Chapter 5 defines the video source to be coded, and develops an analytical model for it. Based on this model suggestions for efficient coding are made. Chapter 6 develops a complete and functioning video coding system, based on the theoretical observations of Chapter 5. Chapter 7 tests this video coding system through comparison to both the existing standards and most recent research proposals.

Finally Chapter 8 concludes the thesis. Here the entire thesis is systematically summarised. The conclusions reached in each chapter are discussed together and the logical flow between chapters is highlighted. A discussion of the algorithm highlighting its strengths and weaknesses and hence potential applications is also given. Based on the weaknesses discussed improvements and possible future work are proposed. Finally the degree of success of this project is determined.

1.3 Executive Summary

The purpose of this project is to produce a low complexity wavelet video compression algorithm.

This work finds that motion estimation and compensation accounts for at least 80% of the computational burden of recent video compression schemes. Difference residual frame coding is proposed as an alternative. It is found that difference frames exhibit strong spatial clustering of significant coefficients. An analysis reveals that source partitioning will allow this clustering to be exploited to gain compression advantage.

An algorithm is proposed that operates by finding the difference between successive frames. By partitioning each difference frame into tiles, and using a fast RD estimator to drive an optimal bit allocation algorithm, an optimal spatial bit distribution is found, thus exploiting the coefficient clustering. Each tile is then wavelet transformed and compressed using the SPIHT algorithm, the embedded stream produced by SPIHT is a key feature allowing the RD optimal bit allocation to be achieved. Finally the output stream of each tile is concatenated, and entropy coded using an arithmetic coder.

A rigorous performance analysis finds that this algorithm executes an order of magnitude faster than H.263+ and MPEG-4. In addition it produces a stream of similar or superior visual quality for local motion scenes. For scenes with global motion the system produces unacceptably low visual quality at low bit rates.

1.4 Context of Project

This thesis represents part of an ongoing research exercise at the University of Natal. Mr Ian McIntosh produced an MScEng thesis in 2001[McIn01] presenting a SPIHT based still image compression system. The wavelet transform and arithmetic coding routines he produced are used in this project. In addition, although a new SPIHT routine was written, Mr McIntosh's work was used to guide the process.

This research project is sponsored by Thales Advanced Engineering and Armscor. It is to fulfil their need for a mobile video compression system that this project was initiated.

Chapter 2 - Rate Distortion Estimation and Optimisation

This chapter will briefly present rate-distortion (henceforth RD) theory. This is a branch of information theory that relates the number of bits used in describing a source, such as an image or video, to the quality of that representation. For instance, in image compression there is the natural relationship that as the compression ratio increases (that is the number of bits decreases), the image quality decreases. The precise nature of this relationship is of interest to us. A detailed knowledge of this relationship is valuable in video compression as it allows the algorithm to concentrate the bit rate in frames, or areas within frames, in a way that will decrease the overall distortion. This process of optimising the algorithm's behaviour based on RD estimation is called RD optimisation, and is also explored below.

The solution is generally framed as a curve that relates the number of bits, or rather the bit-rate (in terms of bits-per-pixel for images), to the distortion introduced to the source.

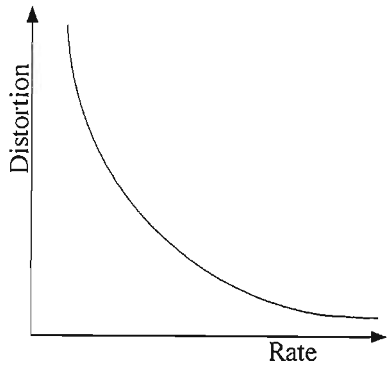


Figure 2-1: General Rate-Distortion (R(D)) Curve

This problem is notoriously hard to solve as the rate-distortion behaviour depends both on the image (the source in RD terminology) and the compression algorithm. Describing either, let alone both simultaneously has proven intractable due to complexity.

The classical approach revolves around the variance of the source. Shannon, who founded this field of study, shows that for a very specific source there exists a closed form analytical relationship between the variance of the original source and the distortion introduced by compressing it to various bit rates. These classical arguments do not consider the impact of the compression algorithm and as a result have proven to be overly conservative, with many existing algorithms exceeding the theoretical performance limits proposed by Shannon. Interestingly Shannon was not working with compression at all. His interest was coding a source (usually vocal) for transmission over a communication channel. His work was aimed at revealing the required channel capacity or bandwidth required to transmit a particular source. It is easy to see how his work has been modified for use in compression, as both revolve around the central idea of determining the 'information content' of a source.

A major problem with the classical treatment is that closed form solutions only exist for specific sources, and assuming that actual images behave similarly to these sources has usually proven fruitless. Section 2.1.1 will give the basics of this theory. Although it is a rich and highly complex area of research, this treatment will not be covered rigorously as it is not used directly in our work.

A more pragmatic approach, termed operational RD estimation, was founded in which the RD curve is populated by points that are actually achievable. This curve is usually referred to as the R(D) curve, as the rate, R, is a function of the required distortion, D. The source in question is compressed to various rates with the algorithm of interest. An interpolation between these points is made and the resulting curve is called the operational R(D) curve. Although accurate, this method is often inappropriate as running the compression algorithm multiple times, for each image (or frame of a video sequence) is computationally exorbitant. Section 2.1.2 will briefly examine this field.

The benefit of using RD estimation to optimise the compression process is well known. Yet in designing compression algorithms, engineers have had to face the choice of poor source representation in order to enable the classical framework, or the computational burden of operational RD estimation. The solutions reached by ISO (the International Standards Organisation) and the ITU (International Telecommunications Union) in their respective MPEG and H.263 standards will be examined in Section 2.1.3.

Recently major progress has been made by Drs He and Mitra at the University of California at Santa Barbara, in solving the RD estimation problem. They have abandoned both the classical treatment and operational RD estimation in favour of a novel approach that they call ρ -domain estimation. This theory parameterises a source in terms of the number of zero or near-zero coefficients that it contains. The fundamental insight is that all compression algorithms treat low value coefficients differently to other coefficients. Thus describing a source in terms of the proportion of low value coefficients (termed ρ), and an algorithm in terms of its ability to handle these low value coefficients should yield insight. This work has proven phenomenally successful and their initial paper on the subject was awarded the Paper of the Year prize by the IEEE Circuits and Systems society. Presenting this theory is the main purpose of this chapter, the previous work is presented mainly to define terms and provide context for this new work.

Having estimated the RD behaviour of a source, it remains to use this estimation to guide the compression algorithm. Many algorithms to be presented in Chapter 3 rely on RD estimation and optimisation. Section 2.2 will present the Lagrange theory of RD optimisation, which is extremely popular and successful. This technique will be incorporated into our algorithm of Chapter 6.

2.1 RD Estimation

This section will present the theoretical basics of RD estimation, from the classical approach of Shannon in Section 2.1.1, to operational RD estimation in Section 2.1.2. Methods of RD estimation that are widely employed through their inclusion in compression standards are discussed in Section 2.1.3. Finally, Section 2.1.4 will discuss the powerful new ρ-domain RD estimation strategy in detail.

2.1.1 Definition and Classical Approach

Shannon presented the first work in this field in 1948 [Shan48] when considering channel coding. In this paper he first defines the relationship between the coding rate for a continuous Gaussian source and the distortion introduced. Later, in 1959 [Shan59] Shannon first coins the phrase "rate distortion function," and develops the now famous Shannon Lower Bound. This paper is focused on discrete sources, whereas his 1948 paper considers continuous sources. This work is fundamental, dealing with abstract sources, although Shannon was mainly interested in channel coding. Video is merely a particular incarnation of an information source, and Shannon's work is directly applicable to this field.

The rate-distortion function represents the minimum number of bits per symbol, the rate, required to represent a source to within a distortion bound D. This function, R(D) is obtained by modelling the source according to a statistical distribution. This section will briefly outline his treatment, following the discussion of [Berg98].

2.1.1.1 The Shannon Formulation

Shannon originally proposed work in channel coding which determines the minimum channel capacity required to transmit a source with a given SNR. Later, he modified this to the problem of source coding. The general formula is [Berg98]:

$$R = W \log_2 (S / D) . \tag{2.1}$$

For this equation to hold, the source must be white Gaussian distributed, bandlimited to $|f| < W$, having a power S. The power spectral density (PSD) must be flat, thus S may be written as $S = S_o W$, where S_0 is the PSD and, as mentioned W is the maximum frequency component of the source. This equation shows that in order to encode such a source with a mean square error (MSE) not exceeding D, R bits per symbol, are required and this is referred to as the rate.

For the case of a Gaussian distributed source, sampled at the Nyquist rate, this equation may be rewritten:

$$R(D) = \max\left[0, \frac{1}{2}\log\left(\frac{\sigma^2}{D}\right)\right] \tag{2.2}$$

As usual σ^2 represents the variance of the source. Expressing the equation in terms of rate, the equation may be rewritten:

$$D(R) = 2^{-2R}\sigma^2 . \tag{2.3}$$

Where the source is not Gaussian distributed, the bound presented above is no longer tight. For the special case where the distribution of the source may be written as the sum of the Gaussian distribution, with variance Q_0, and another distribution, the following bound may be written [Berg98]:

$$R(D) = \frac{1}{2}\log\left(\frac{Q_0}{D}\right), \qquad 0 < D < D^*. \qquad (2.4)$$

D^* represents the highest distortion for which the sum of distributions represent the original distribution.

In general it may be shown [Proa00] that for any memoryless continuous source with zero mean, and variance σ^2, equation (2.2) represents a lower limit on the rate.

For other distributions there is not a closed form solution for R(D) [Orte98] and various approximations and numerical methods are employed.

2.1.1.2 Summary of Shannon's RD Estimation Theory

This section has introduced the Shannon approach to source coding, in very brief terms. It is unfortunate to provide such a short treatment to this fascinating field, but it is not directly relevant to our work.

There are two main problems that arise from this theory. The first is modelling error; the models assumed are not necessarily representative of real data sources. In many cases assuming the source to be distributed according to the Gaussian function is a gross simplification. Using such a model as the basis of a source estimation will lead to overly conservative R(D) functions.

The second problem is that this RD theory is not constructive. No indication is given by the theory of how to code a source such that the output stream approaches the R(D) bound as given by the theory.

A considerable volume of research has been directed at solving these problems. However, the intricate mathematics in which this theory is steeped prevents it discussion here or its practical application in most cases.

2.1.2 Operational Rate-Distortion

Due to the problems with the analytic approach to RD estimation, an approach known as operational rate distortion theory has been developed [Orte98]. Under this school, achievable points in the R(D) curve are obtained by coding a representative source (or the actual source) with the algorithm in question, at different rates, and noting the resultant distortion. A curve, such as the following may be generated from these test points.

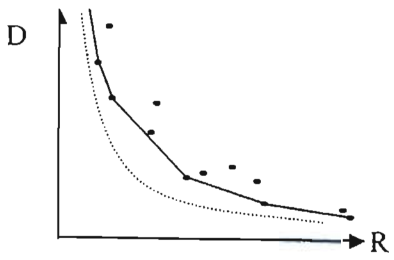


Figure 2-2: Operational R(D) Curve

Knowing that the R(D) is strictly monotonically nonincreasing [Berg98], an interpolation between the lower points on the curve may be made, as in Figure 2-2. The dotted line in this figure represents the theoretical R(D) curve, which is unknown. Although the above curve uses a linear interpolation, more complex spline based approaches are common.

Although this approach provides a numerical curve on which RD decisions may be made, the computational burden of populating the curve with R-D pairs is in many cases unacceptable. Thus the need still remains for an analytical solution to generate the expected RD behaviour of a coding system.

2.1.3 Methods used in Standards

Section 2.1.1 presented the classical RD approach based on Shannon's Theorems. It was found that both the simple statistical models employed, and the lack of a general solution yield poor estimation performance for complex data. Section 2.1.2 demonstrated that an R(D) curve may be obtained empirically, but that the computation involved is burdensome.

This section will briefly outline the RD models currently employed in common video encoding standards. Unfortunately RD estimation is usually only used in RD control for video. Thus the published results are for rate control, and whether the behaviour is attributable to the RD estimation or subsequent rate control is hard to discern.

Rate control is first explained for MPEG2, which is the most widely used standard at this time, being employed for DVD encoding as well as digital television. H.263+ rate control then explained. H.263+ is a widely adopted internet videoconferencing method. Finally the RD model used in the extremely recent MPEG4 codec is presented.

2.1.3.1 MPEG2 (TM5)

MPEG2[MPG93] employs an extremely simple RD model:

$$R(q)\cdot D(q)=k\ , \tag{2.5}$$

where q is the quantisation parameter (such as the quantiser stepsize in the case of uniform threshold quantisation), and k is a constant, estimated from the data. It is evident that this technique will perform poorly due to over simplification of the R(D) curve.

2.1.3.2 H.263+ MPEG4

Ribas-Corbera and Lei present a representative RD estimation and rate control algorithm in [Corb99]. Their algorithm has been incorporated into TMN8 (Test Model 8) of the H.263+ video compression codec, and VM8 (Verification Model 8) of MPEG-4. It may thus be taken as a representative of the state of the art technique.

As do most previous algorithms, TMN8 considers both the rate and distortion as functions of the quantisation parameter, q. Thus two functions are estimated, R(q) and D(q), where in this case q represents the stepsize of a uniform quantiser. As this algorithm is used for RD estimation of motion compensated frames, each frame is modelled as a Laplacian distribution, with a variance of σ^2.

The entropy of each frame is modelled as:

$$H(q) = \begin{cases} \frac{1}{2}\log_2\left(2e^2\frac{\sigma^2}{q^2}\right), & \frac{\sigma^2}{q^2} > \frac{1}{2e} \\ \frac{e}{\log_e 2}\frac{\sigma^2}{q^2}, & \frac{\sigma^2}{q^2} \leq \frac{1}{2e} \end{cases} \tag{2.6}$$

based on work in [Corb96]. The upper equation is the high rate approximation; when the quantization parameter, q, is small, the rate is large. This is obtained by equating the differential entropy of the source, with the differential entropy of a Laplacian distribution. The low rate approximation is simply a linearization, in terms of the variance, of the upper equation.

By equating the rate to the entropy, which is an approximation, the low rate function may be written:

$$\tilde{R}(q) = \frac{e}{\log_e 2}\frac{\sigma^2}{q^2} \approx K\frac{V^2}{q^2} \tag{2.7}$$

where V is the mean value of the variance, taken over several frames, and K is also estimated over several frames. K is not taken to be $1/\log_e 2$, which would make the second approximation exact, for the Laplacian case. However, as the Laplacian assumption is inexact, the authors found an adaptive K yields superior performance.

The distortion is also simply modelled as

$$D_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\frac{q_i^2}{12}, \tag{2.8}$$

for each macroblock of the frame, where N is the number of coefficients in the macroblock and q_i is the quantiser stepsize. The 12 in the denominator is a consequence of using uniform coefficient quantization [Goya01].

There are several problems related to this method. Both the rate and distortion models are simplified. Furthermore due estimation of parameters K and V,. based on previous frames, scene changes cause large errors [He01].

2.1.3.3 MPEG4 (VM8)

Lee *et al* in [Lee00] show a frame level rate control algorithm that is also included in the MPEG 4, VM8 specification. Modelling the motion predicted frames as Laplacian distributed, as in TMN8 above gives:

$$P(x) = \frac{1}{\sigma\sqrt{2}} e^{\frac{-\sqrt{2}|x|}{\sigma}}, \tag{2.9}$$

which allows the Shannon bound to be rewritten:

$$R(D) = \log_2\left(\frac{\sigma}{\sqrt{2} \cdot D}\right), \tag{2.10}$$

where $0 < \left(D = |x - \tilde{x}|\right) < \frac{\sigma}{\sqrt{2}}$. The limit on the distortion having been explained in 2.1.1.1.

Now expanding R(D) into a Taylor series yields the following formula:

$$\begin{aligned} R(D) &= \left(\frac{\sigma}{\sqrt{2}D} - 1\right) - \frac{1}{2}\left(\frac{\sigma}{\sqrt{2}D} - 1\right)^2 + R_3(D) \\ &= -\frac{3}{2} + \frac{2\sigma}{\sqrt{2}} D^{-1} - \frac{\sigma^2}{4} D^{-2} + R_3(D) \end{aligned} \tag{2.11}$$

As *D α q* [Hang97] ([Hang97] uses the symbol *Q* for *q*) when defining D as a difference-, rather than the usual squared-metric (that is a Manhatten rather than Euclidean error measure), the R(D) may be rewritten in terms of the quantisation stepsize:

$$R(q) \approx a_1 \cdot q^{-1} + a_2 \cdot q^{-2}, \tag{2.12}$$

by retaining only the distortion terms in (2.11).

The parameters a_1 and a_2 are obtained by linear regression. Operating under the assumption that the RD behaviour of subsequent frames will be similar, the preceding frame's values of a_1 and a_2 are used to estimate a_1 and a_2 of the current frame during coding.

As with TMN8 estimating the parameters a_1 and a_2, based on the previous frame, introduces errors at scene changes.

2.1.3.4 Summary of Standards Methods

As the previous sections show, the employed standards all share several similar features. The rate and distortion are described separately, and as functions of the quantisation parameter, q. Operational RD theory is not employed due to the high computational burden incurred. Although the employed models may be loosely based on statistical models in their theoretical development, in practice the R and D behaviour is obtained by curve fitting techniques. It has been found [He01] [Orte98] that due to the many assumptions, and lack of fundamental agreement with theory, these RD estimators suffer from large inaccuracies.

The performance of these algorithms is displayed in Section 2.1.8, where they are compared to a new technique.

2.1.4 ρ-Domain RD Estimation

From the previous discussion it is clear that current RD estimation methods are unsatisfactory. Statistical modelling presents poor source representation for complex sources such as video, and operational RD theory is computationally intensive.

Recent work by He and Mitra in [He01] presents a novel departure in the field of RD estimation, the technique presented offers both superior estimation accuracy, and computational ease. The method is based on neither explicit statistical models, nor an operational RD approach; the estimate is based instead on numerical methods and regression. The remainder of this section is heavily based on [He01] and [He01,2].

2.1.4.1 Premise

The RD estimation models presented to this point are written in terms of the quantisation parameter, q, which is usually the step size of the quantiser. He and Mitra present a new variable, ρ, which is defined as the fraction of the total coefficients that have been quantized to zero, in a particular compressed representation.

The RD characteristics of the source and codec are considered independently, and modelled in terms of ρ. In addition, fast methods of developing these models are presented which allow this method to be implemented in a real time scenario.

The original work, [He01], presents a unified approach that may be utilized for any codec, however here we will discuss only the application to wavelet based codecs. This original work only considers still image wavelet coding, and DCT video coding. In Chapter 6 we will consider applying this work to wavelet video coding.

2.1.5 The ρ- Domain

Lossy compression is always achieved by some form of quantization. Specifically, in zerotree quantization schemes (Section 3.), large spatial regions of coefficients are quantized to zero if they fall below a certain threshold, Δ. For instance SPIHT (to be discussed in Section 3.3.4) outputs an embedded stream, where at each significance level, every coefficient may be considered to be quantized according to:

$$I(x) = \begin{cases} 0, & |x| < \Delta \\ \left\lceil \dfrac{x-\Delta}{\Delta} \right\rceil, & x > +\Delta \\ \left\lfloor \dfrac{x+\Delta}{\Delta} \right\rfloor, & x < -\Delta \end{cases} \qquad (2.13)$$

If this form of quantisation is used, the ρ may be written as a function of the significance level:

$$\rho(\Delta) = \frac{1}{M} \int_{-\Delta}^{\Delta} D(x)dx \quad , \tag{2.14}$$

where M is the total number of coefficients, and $D(x)$ is the distribution of these coefficients.

Using this parameter, ρ, it is possible to conceive the curves $R(\rho)$ and $D(\rho)$. The consideration of these curves is termed ρ-domain analysis.

2.1.6 ρ-Domain Analysis

The RD characteristics of any given encoded stream are dependant on the input source and the compression algorithm used. Thus a characterization of both needs to be made independently. A function that relates the overall RD behaviour to each of these characterizations then needs to be formed. This section will examine how these tasks are accomplished.

The first section to follow, 2.1.6.1, will present the high level rate and distortion models used in the method. Section 2.1.6.2 will then explore the rate modelling of the source, in detail. Section 2.1.6.3 will consider the rate modelling of the compression algorithm. Both models are combined in Section 2.1.6.4, which considers how the rate model is implemented in real time. The distortion model is far simpler and the discussion limited to Section 2.1.6.1.

2.1.6.1 High Level Rate and Distortion Modelling

He and Mitra consider a linear rate equation:

$$R(\rho) = A(\rho)\cdot Q_{nz}(\rho) + B(\rho)\cdot Q_z(\rho) + C(\rho) \tag{2.15}$$

The two equations $Q_z(\rho)$ and $Q_{nz}(\rho)$ are used to model the source. $Q_z(\rho)$ is an estimate of the number of bits required for a raw representation of the zero quantized bits. $Q_{nz}(\rho)$ is an estimate of the raw bit rate of the non-zero quantized bits. Section 2.1.6.2 will describe these functions in more detail.

The three parameters, $A(\rho), B(\rho)$ and $C(\rho)$ model the coding algorithm. $A(\rho)$ and $B(\rho)$ model the algorithm's ability to compress the non-zero and zero coefficients respectively, and $C(\rho)$ models the rate overhead of the algorithm. These parameters will be explored in Section 2.1.6.3.

The distortion modelling is particularly simple in the ρ-domain, as the information may be obtained directly from the source distribution. In [Jack03] the proposed method is:

$$D(\Delta) = \frac{2}{M}\left(\int_{0}^{\Delta} x^2 \cdot D(x)dx + \int_{\Delta}^{\infty} \Delta^2 dx \right)^{1/2} , \tag{2.16}$$

where ρ and Δ are related by

$$\rho(\Delta) = \frac{2}{M}\int_{0}^{\Delta} D(x)dx \qquad (2.17)$$

In practice Δ may found from ρ readily using equation (2.17) numerically, from which (2.16) may be used to calculate the MSE distortion.

2.1.6.2 Source Rate Model

2.1.6.2.1 First Principles

This section will present the functions $Q_z(\rho)$ and $Q_{nz}(\rho)$. Initially these functions are evaluated from the theory presented; 'first principles'. Thereafter fast methods for the evaluation of each will be presented.

$Q_{nz}^{'}$ is the number of bits required to represent, in sign-magnitude notation, all the non-negative quantized coefficients.

$$Q_{nz} = \sum_{\forall x \neq 0} \lfloor \log_2 I(x) \rfloor + 2 \qquad (2.18)$$

Hence Q_{nz} may be defined as the per coefficient value:

$$Q_{nz} = \frac{1}{N} Q_{nz}^{'} \qquad (2.19)$$

where N is the number of non-zero coefficients.

The zero coefficients are considered slightly differently. Rather than count the number of bits required to encode the number of zero quantised coefficients, which would simply be the number of such coefficients, the number of bits required to encode the runlength of each run of zeroes is calculated by

$$Q_z = \frac{1}{M} \sum_{zeroruns} \lfloor \log_2(runlength) \rfloor + 2. \qquad (2.20)$$

As the number of zero and non-zero quantized coefficients is dependant on the quantization threshold used, the curves $Q_{nz}(\Delta)$ and $Q_z(\Delta)$ may be developed, and hence $Q_{nz}(\rho)$ and $Q_z(\rho)$ may be found using equation (2.17). These measurements characterize the source.

2.1.6.2.2 Results

In order to examine the properties of $Q_{nz}(\rho)$ and $Q_z(\rho)$ in the ρ-domain, these functions were evaluated for the set of images shown in Figure 2-3.

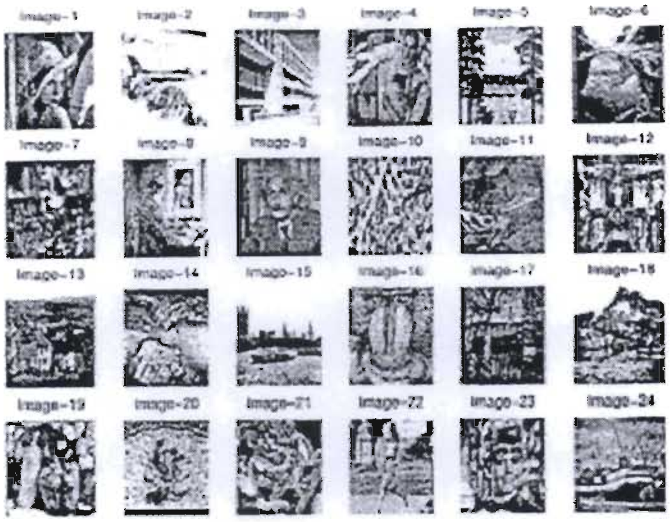


Figure 2-3: Test Image Set for Qz and Qnz[He01]

The wavelet transform was performed on each image, followed by uniform threshold quantisation at various levels. The curves found for $Q_{nz}(\rho)$ and $Q_z(\rho)$ are shown in Figure 2-4.

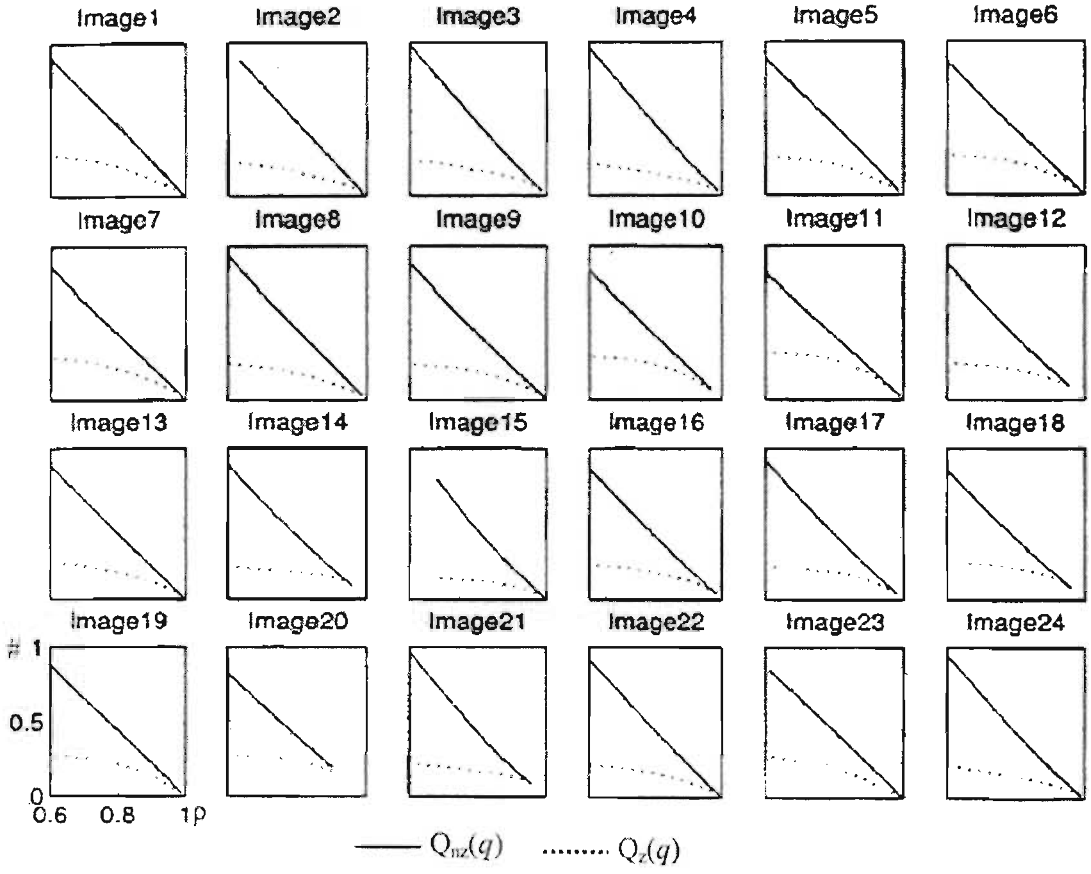


Figure 2-4: Qnz and Qz for Test Set[He01]

These curves are striking in their simplicity. Each curve is drawn to identical axes, where the x-axis is the value of ρ and the y-axis the values taken on by $Q_{nz}(\rho)$ and $Q_z(\rho)$. $Q_{nz}(\rho)$ is the solid line, and $Q_z(\rho)$ the dotted line. In general terms the curves are sensible. Both curves pass through the (1,0) point. An image where 0 occurs with probability 1 has entropy of

$$H = \sum_{i=0}^{N-1} p_i \log(p_i) = \sum_{i=0}^{N-1} 1.0 = 0 , \tag{2.21}$$

and accordingly requires no bits to describe. An explanation of the behaviour of each curve will be given shortly.

For comparison He and Mitra produce curves of $Q_{nz}(q)$ and $Q_z(q)$:

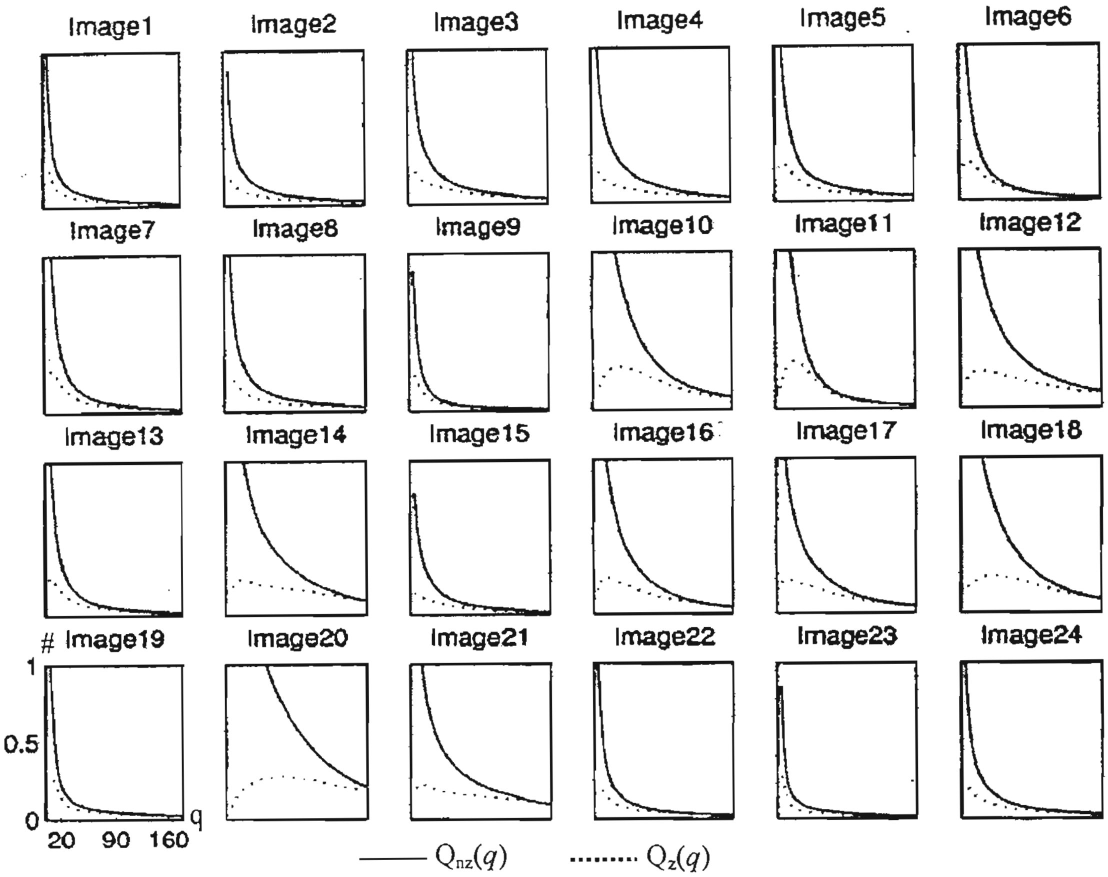


Figure 2-5: Qnz and Qz parameterised by q [He01]

The nonlinearity and diverse nature of the curves shows the power of the ρ parameter. Characterising an image in terms of ρ yields much simpler curves that should be more predictable, thus aiding greatly in the RD estimation process.

This behaviour in the ρ-domain may be explained by the behaviour of zerotree coders (indeed most coders). These coders treat insignificant coefficients very differently to significant ones. The premise of EZW [Shap93] is that the specific behaviour of insignificant coefficients allows them to be predicted with some accuracy. The entire zerotree algorithm is based on this observation. The ρ -domain method acknowledges this separate treatment in the compression algorithm through the use of $Q_{nz}(\rho)$ and $Q_z(\rho)$.

2.1.6.2.3 Fast Estimation of Qnz(ρ)

As Figure 2-4 clearly shows, $Q_{nz}(\rho)$ approximates a straight line. He and Mitra justify this very distinctive behaviour as follows. Modelling the transformed coefficients as a generalised Gaussian distribution, as is common practice [Lopr97], the probability of x is:

$$p(x) = \left[\frac{\upsilon \cdot \eta(\upsilon,\sigma)}{2\Gamma\left(\frac{1}{\upsilon}\right)}\right] \cdot e^{-[\eta(\upsilon,\sigma)|x|]^{\upsilon}} \tag{2.22}$$

where the shape parameter is given by

$$\eta(\upsilon,\sigma) = \sigma^{-1}\left[\frac{\Gamma\left(\frac{3}{\upsilon}\right)}{\Gamma\left(\frac{1}{\upsilon}\right)}\right]^{1/2} \qquad 1 \le \upsilon \le 2\,. \tag{2.23}$$

Writing equation (2.17) using this distribution:

$$p(\Delta) = \frac{2}{M}\int_{0}^{\Delta} p(x)dx\,, \tag{2.24}$$

and evaluating it numerically yields the following results.

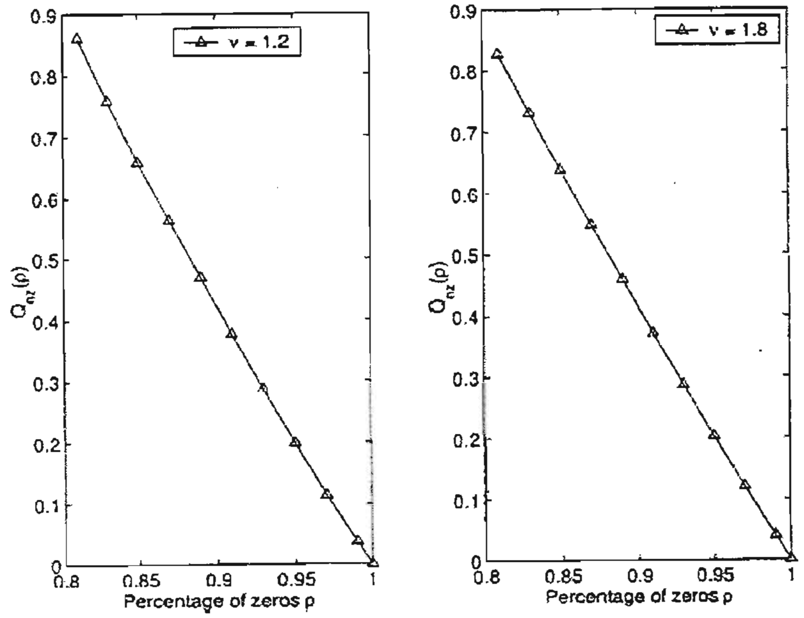


Figure 2-6: Qnz(p) plotted for a GGD [He01]

This numerical evidence provides justification for the linearity of $Q_{nz}(\rho)$.

Thus $Q_{nz}(\rho)$ is modelled as a straight line passing through the point (1,0). For this reason a fast estimation consists of merely evaluating $Q_{nz}(\rho)$ for any value of $\rho \neq 1$. Any subsequent points are found by interpolation.

The gradient of this line is given the parameter κ, which is used in the estimation of $Q_z(\rho)$.

2.1.6.2.4 Fast Estimation of Qz(p)

The behaviour of $Q_z(\rho)$ is not as simple as $Q_{nz}(\rho)$, however He and Mitra present a numerical method, based on regression that provides a fast estimation framework.

Considering κ to define the curve of $Q_{nz}(\rho)$ He and Mitra explore a possible correlation between the curves of $Q_{nz}(\rho)$ and $Q_z(\rho)$. Using the test image set in Figure 2-3, $Q_z(\rho)$ is evaluated for $\rho = \{0.70; 0.75; 0.80; 0.85; 0.90; 0.95\}$. The following curves are produced by plotting $(\kappa, Q_z(\rho))$ for each of the test images. Each graph is for a particular value of ρ.

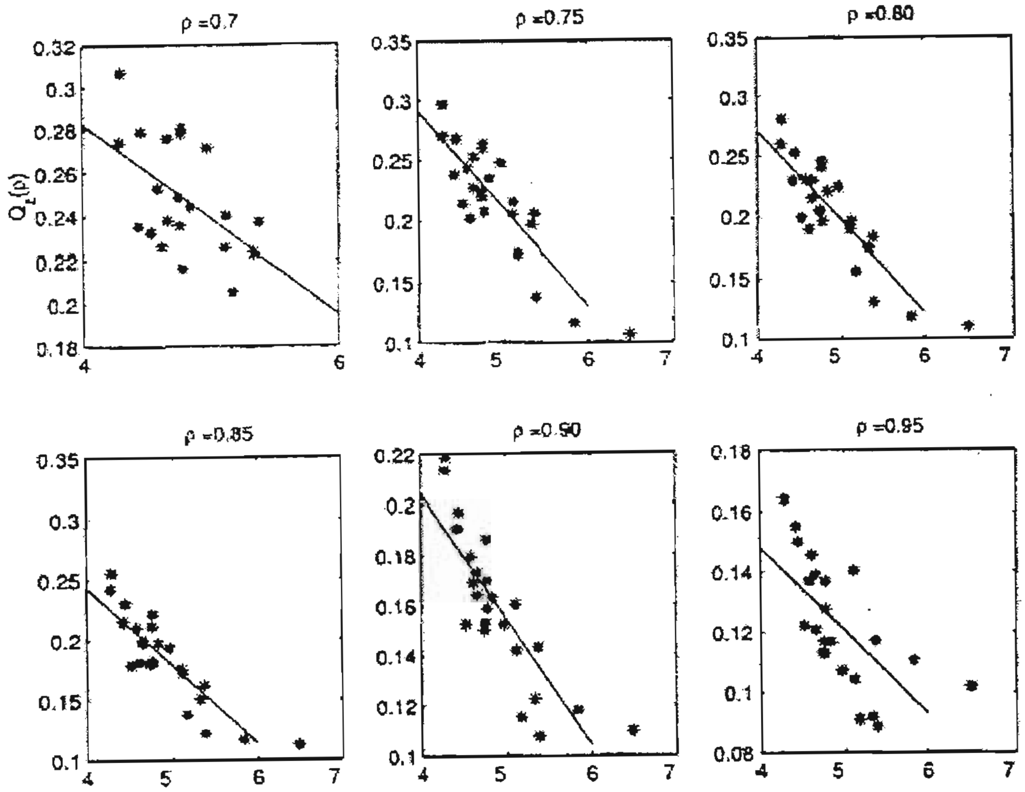


Figure 2-7: Correlation between Qz(p) and Qnz(p) [He01]

The strong correlation is evident. Thus the following linear model for $Q_z(\rho)$ is proposed:

$$Q_z(\rho_i) = A_i\kappa + B_i\,, \tag{2.25}$$

where the parameters A_i and B_i are found using an offline statistical regression. This process, and further results, will be detailed in Chapter 5.

2.1.6.3 Compression Algorithm Rate Model

The rate required to represent an image depends both on the image and the algorithm used to compress it. This is expressed in the rate equation (2.15):

$$R(\rho) = A(\rho)\cdot Q_{nz}(\rho) + B(\rho)\cdot Q_z(\rho) + C(\rho) \tag{2.15}$$

As the previous section has shown, $Q_z(\rho)$ and $Q_{nz}(\rho)$ are used to model the source. This section will show how $A(\rho), B(\rho)$ and $C(\rho)$ model the coding algorithm.

The estimation technique is based on offline operational rate-distortion techniques. The coding algorithm to be characterised is run several times at different output rates, for each image in the set, to generate an operational rate distortion curve.

Equation (2.15) is then fitted to this curve, again using linear regression, and the values of $Q_z(\rho)$ and $Q_{nz}(\rho)$ found previously. In this way the coefficients $A(\rho), B(\rho)$ and $C(\rho)$ are found.

If the test set is correctly chosen, then these coefficients will be fixed after this point. Thus the regression described is performed offline, and the coefficients simply stored in the algorithm.

2.1.6.4 Rate Regulation

Based on empirical evidence, He and Mitra find that the function $R(\rho)$ is approximately linear. However, due to error in obtaining any of the values or curves, or unusual circumstances, $R(\rho)$ obtained by (2.15) may not be linear. By using a least mean square technique, He and Mitra find the best fit linear curve for the points estimated using (2.15). The slope of this curve, by linear regression over six estimated points is given by [He01]:

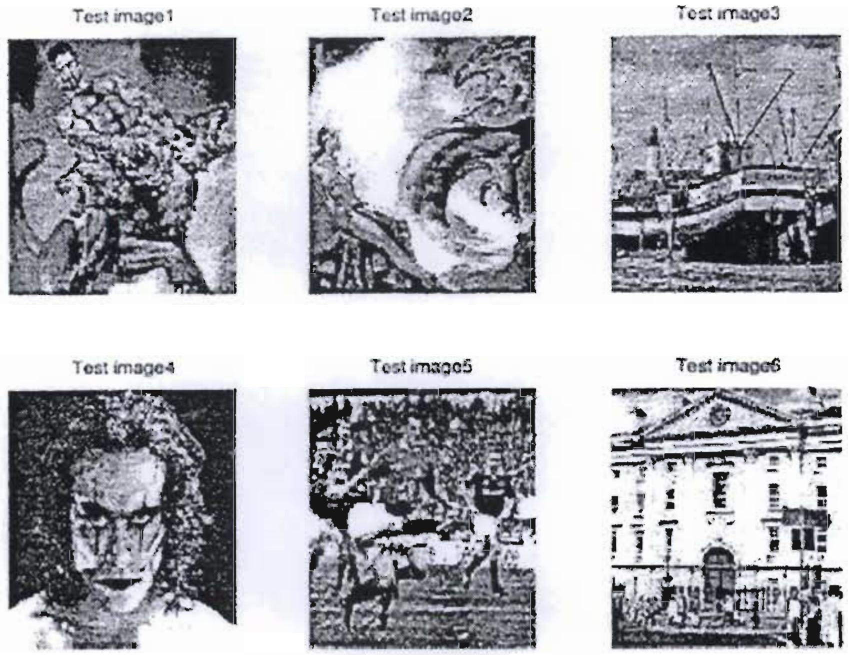
$$\theta = \frac{\sum_{i=0}^{5} \rho_i \sum_{i=0}^{5} R(\rho_i) - 6\sum_{i=0}^{5} \rho_i \cdot R(\rho_i)}{6\sum_{i=0}^{5} \rho_i^2 - \left(\sum_{i=0}^{5} \rho_i\right)^2}, \tag{2.26}$$

and hence a best fit linear curve may is described by [He01]:

$$R(\rho) = \frac{1}{6}\sum_{i=0}^{5} R(\rho_i) + \theta \cdot \left(\rho - \frac{1}{6}\sum_{i=0}^{5} \rho_i\right). \tag{2.27}$$

2.1.7 Results

He and Mitra [He01] present results for the RD estimation scheme. The rate is found using equation (2.27), and although the equation used to find the distortion is not given, it is assumed to be of the form of (2.16). The following validation image set, which is different from the set used to perform the regressions to find the coefficient, is used.



Test image1 Test image2 Test image3

Test image4 Test image5 Test image6

Figure 2-8: Validation Set [He01]

The wavelet transform of each image is found, and compressed using two techniques, Stack Run Coding, (SR) (Section 3.3.7), and Set Partitioning in Hierarchical Trees (SPIHT) (Section 3.3.4). The actual RD behaviour that results is compared with that estimated using the RD estimation scheme.

The SR results are:

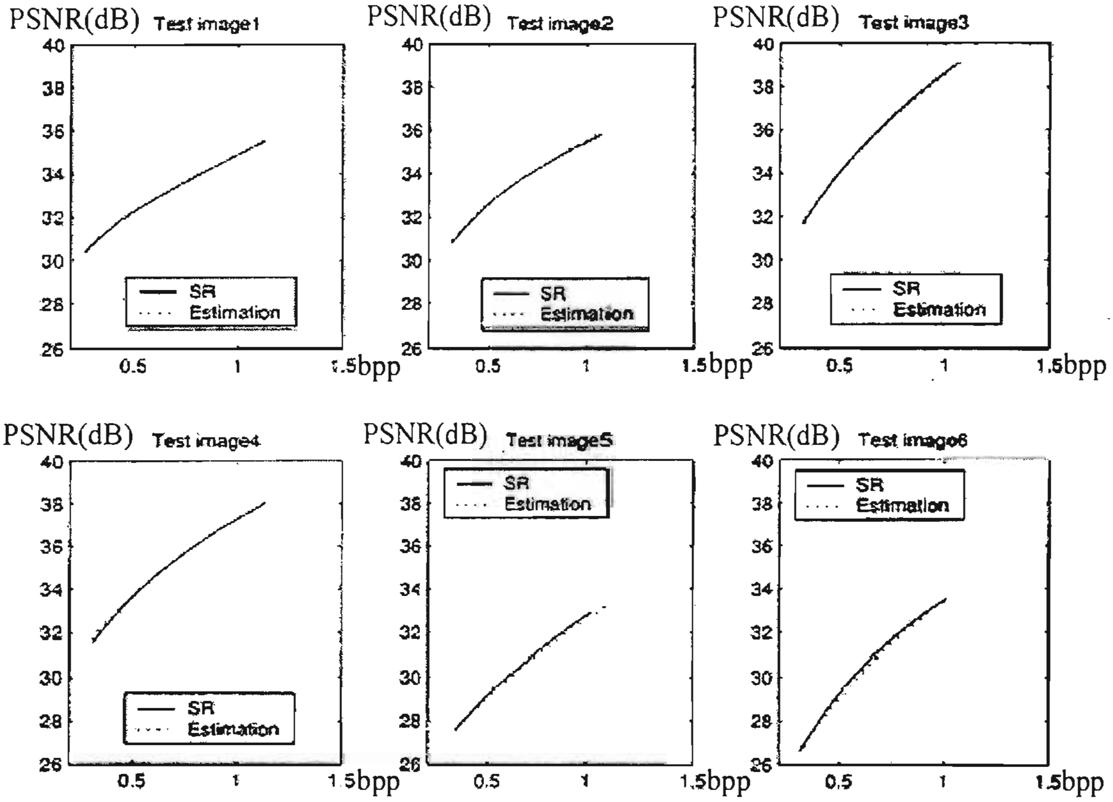


Figure 2-9: SR RD Estimation Performance

In these curves the x-axis represent the rate in bits per pixel (bpp), and the y-axis the quality (inverse of distortion, measured in dB, as explained in (Section 3.1)). The solid lines are the behaviour of the algorithm, and the dotted lines are the RD estimate. The accuracy of the estimate is extraordinary.

Similar results are found using SPIHT:

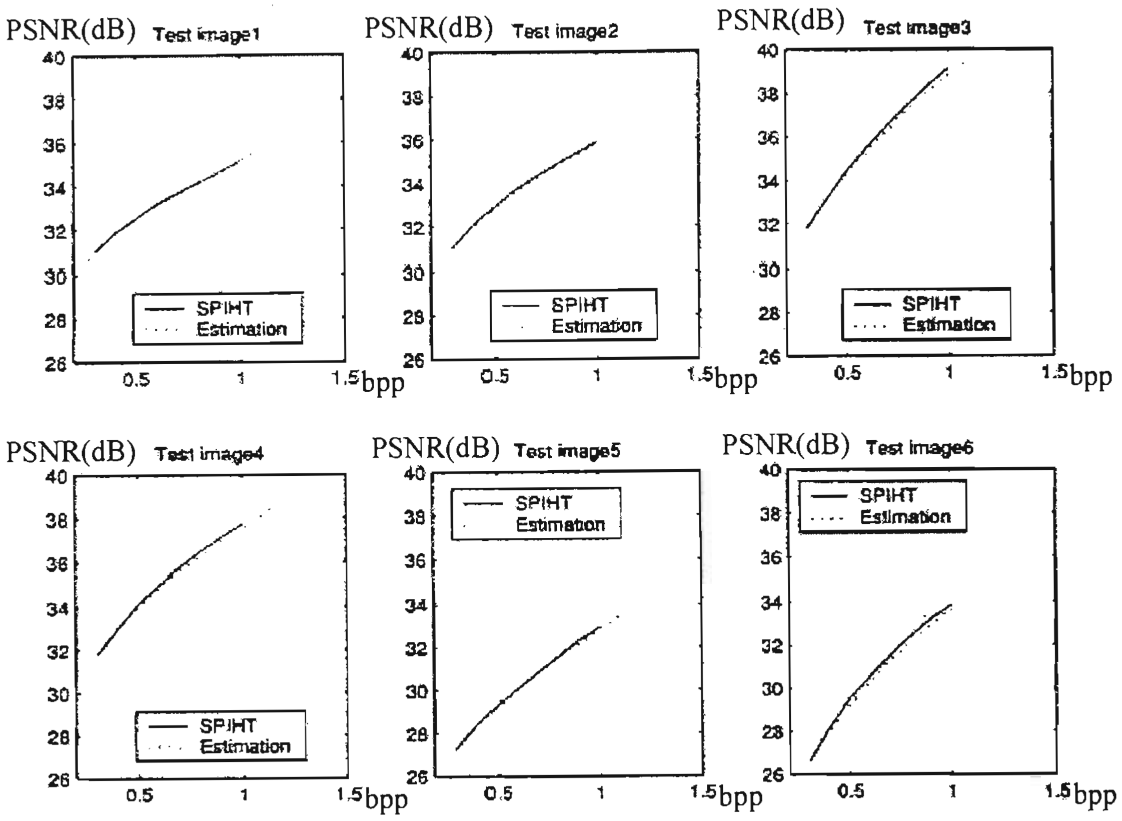


Figure 2-10: SPIHT RD Estimate Performance

The results are equally striking.

2.1.8 Rate Control

As mentioned in 2.1.3 RD estimation is often used in video coding to perform rate control. Based on the RD estimation, a rate target is set for each frame, in order to meet some RD criterion. This target rate specifies the quantisation parameter with which the frame is encoded. After encoding the frame any error in the RD estimation will cause the achieved rate to differ from the target rate.

The following curves show the relative error between the target and achieved rate for the TM, VM8 and ρ-domain methods. Although not a direct method, these graphs do give an indication of the RD performance of each method.

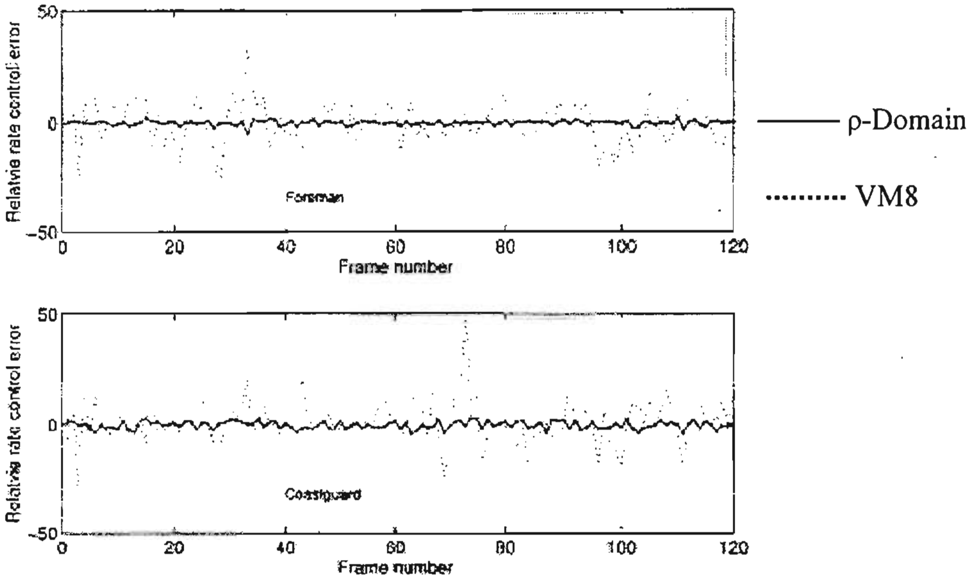


Figure 2-11: TM5 Rate Control Error [He01]

The rate control error is expressed as a percentage of the target rate. The upper graph is for the 'Foreman' sequence and the lower graph is for the 'Coastguard' sequence. The relatively large rate control error of the MPEG-2 TM5 algorithm is sharply contrasted by the accuracy of the proposed ρ-domain method.

The rate control performance of the MPEG-4 VM8 algorithm is shown in Figure 2-12 for the 'News' sequence.

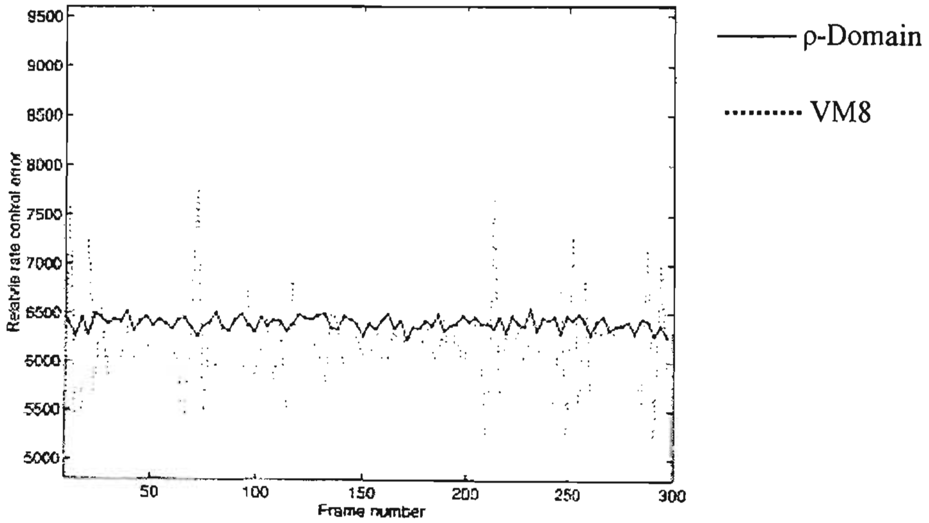


Figure 2-12: MPEG4 Rate Control [He01]

Once more the superior performance of the ρ-domain method is apparent.

2.1.9 Summary

The ρ -domain RD estimation algorithm has been fully developed. It is seen to rely on implicit statistical modelling of the source and compression algorithm through multiple regression stages. This is opposed to the general practice that uses explicit statistical modelling, or operation RD techniques.

The performance of the scheme is superior to any other technique presented in the literature [He01]. In addition to this the technique is very fast online, consisting of finding the distribution of the transform coefficients, followed by simple mathematical operations. All the statistical characterisation is performed offline. This is possible due to the predictable behaviour of both the source and algorithm in the ρ -domain.

The importance of this technique was recognised by the IEEE Circuits and Systems Society which awarded the paper, [He01], the prize for Best Paper in 2003.

The dramatic performance of this scheme has motivated for its use in the work presented in this thesis. Chapter 5 will present an implementation of the algorithm, further results, and modification for use in a wavelet video coder, as opposed to still image wavelet coders presented above.

2.2 RD Optimisation

Section 2.1 has discussed various techniques of determining the RD behaviour of a source. This section will lead on from this discussion, to explore means of using this RD estimation to guide the compression process. Again, there are many more RD optimisation strategies in the literature than may be discussed in this limited space, thus only the most popular and widely used method, based on the theory of Lagrange multipliers will be discussed. Chapters 3 and 4 lend credence to this decision through the number of existing image and video compression algorithms that rely on Lagrange's theory. Finally, the method developed in Chapter 6 is also based on Lagrange multipliers, thus this section is very detailed.

Section 2.2.1 develops the theory from a purely mathematical perspective. The compression engineering interest in the theory is introduced in Section 2.2.2. Finally the joint optimisation of multiple units is considered in Section 2.2.3. This method of optimisation is used in the proposed method of Chapter 6.

2.2.1 Theory of Lagrange Multipliers

This theory is fundamental in differential calculus, and is used to optimise a function given another constraining function. The general form of this optimisation is given below, following the development in [Thom96].

Given two differentiable equations, $f(\cdot)$ and $g(\cdot)$, the local maximum or minimum of f subject to $g = 0$ is given by

$$\nabla f = \lambda \nabla g\,, \tag{2.28}$$

where λ is a scalar quantity, known as the Lagrange Multiplier, and ∇ is the gradient operator. Restricting the discussion to the two dimensional case (although the theory is general), the gradient of $f(x,y)$ at the point (x_0, y_0) is defined as the vector

$$\nabla f = \left.\frac{\partial f}{\partial x}\right|_{(x_0,y_0)} \underline{\mathrm{i}} + \left.\frac{\partial f}{\partial y}\right|_{(x_0,y_0)} \underline{\mathrm{j}} . \tag{2.29}$$

An important property of the gradient vector is that it is the vector normal to $f\left(x_{0,}y_0\right)$.

Consider a parameterisation of the constraint:

$$g(x,y) = x(t)\underline{\mathrm{i}} + y(t)\underline{\mathrm{j}} . \tag{2.30}$$

This parameterisation is substituted into f, in order to evaluate the function only on the constraint. The derivative of f with respect to the parameter t is given by:

$$\begin{aligned} \frac{\partial f}{\partial t} &= \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \\ &= \left[\frac{\partial f}{\partial x}\underline{\mathrm{i}} + \frac{\partial f}{\partial y}\underline{\mathrm{j}}\right] \bullet \left[\frac{dx}{dt}\underline{\mathrm{i}} + \frac{dy}{dt}\underline{\mathrm{j}}\right], \\ &= \nabla f \bullet \underline{v} \end{aligned} \tag{2.31}$$

where, by standard notation $\underline{v}$ is the velocity vector of the function g. At any local maximum or minimum this derivative will be equal to zero:

$$\frac{df}{dt} = \nabla f \bullet \underline{v} = 0 . \tag{2.32}$$

Equation (2.32) states that at the local maximum or minimum of f, on the curve g, the gradient vector of f, ∇f, will be orthogonal to the velocity vector of g, $\underline{v}$.

By their respective definitions, a curve's velocity vector is tangential, and gradient vector is normal to that curve at any given point. Thus they are orthogonal and the gradient vector of f, ∇f, will be parallel to the gradient vector of g, ∇g, at optimality. This is a statement of equation (2.28).

Alternatively, equation (2.28) may be interpreted as the condition at which the two curves, f and g, share a common tangent.

2.2.2 Application of Lagrange Multipliers to RD Optimisation

Consider a typical R(D) curve, as shown in Figure 2-13 below.

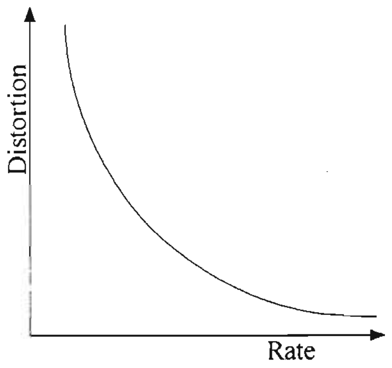


Figure 2-13: Typical R(D) Curve

In the Lagrange terminology, this function may be viewed as the constraint, g, as it represents the performance that the coding system is capable of producing. The function to be optimised, f, is referred to as the cost function in the literature, and written as $J = D + \lambda R$. The situation may be cast as a Lagrange optimisation problem thus:

$$\begin{aligned} &g : D(R) \\ &f : J = D + \lambda R \end{aligned} \qquad (2.33)$$

The objective of this optimisation problem is to minimise the cost function, J.

Consider the case of a fixed Lagrange Multiplier, λ. Although it is possible to fix any of the other parameters and perform the optimisation, the utility of fixing λ will become clear in the following section on optimisation over multiple units.

The possible cost functions available by varying the cost, J, are shown alongside the R(D) curve below in Figure 2-14.

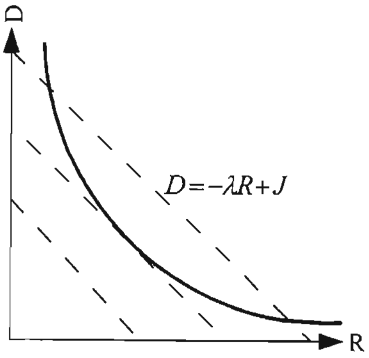


Figure 2-14: RD Cost Functions

Now from the final paragraph of the Lagrange Theory section, the point on the R(D) curve which minimises the cost function is that point at which the two curves are tangential. The gradient of the cost function is $-\lambda$ at all points, and by the RD theory given in Section 2.1.2, the

R(D) curve is strictly non-increasing. Thus for a given point on the R(D) curve there is one cost function curve that is tangential.

The cost , *J*, of the cost function, is represented geometrically as the D-intercept of the curve. By observation, the minimum cost function is that one which is tangential to the R(D) curve.

This section showed that minimising the cost function corresponds to finding the point on the R(D) curve with gradient - λ . This λ is equal to the gradient of the tangent to the R(D) curve, that is

$$\lambda = \frac{d}{dR} D(R) \Big|_{(x_0, y_0)} \qquad (2.34)$$

Thus if D(R) is linearised about a point (x_0, y_0) , λ represents the rate of decrease of D with respect to an increase in R.

2.2.3 Optimization of Multiple Units

In most cases image compression systems consist of multiple phases. These different coding units should ideally all operate at RD optimality, that is, the system as a whole should introduce the smallest distortion possible for a given rate.

In order to illustrate the process, a commonly encountered problem will be used as an example. Several image and video compression systems (examples exist in Chapters 2, 3, 5 and 6), segment each frame into a number of tiles, which are processed independently. It is desirable that each of these tiles operate such that the overall distortion is minimized, given a total rate per frame.

A method of achieving this optimality is to estimate the RD behaviour of each tile under the given compression scheme. Based on these curves, it is possible to allocate the rate between the tiles in a globally optimal fashion, by ensuring that each tile is operating at the same value of λ . Section 2.2.2 has shown that given a fixed value of λ , a point on the R(D) curve may be found that minimises the cost function, this may be considered a local optimality. It would seem logical that should the same cost function be minimised over each tile, it would be minimised over the entire image. The following development tests this hypothesis.

Proof of the global optimality of this situation is offered by the following development [Jack03]. Consider an image segmented into N tiles, then:

$$\begin{aligned} Dtot_{\mathrm{nom}}(R_{tot}) &= \sum_{n=0}^{N-1} D_n(R_{o,n}) \\ \mathrm{R}_{\mathrm{tot}} &= \sum_{n=0}^{N-1} R_{o,n} = R_{budget} \end{aligned} \quad , \qquad (2.35)$$

and two blocks operating at nominal rates $\mathrm{R}_{0,1}$ and $\mathrm{R}_{0,2}$ respectively, where,

$$\left|{}^{dD_1}\!\!/_{DR}\right|_{R_1=R_{01}} > \left|{}^{dD_2}\!\!/_{DR}\right|_{R_2=R_{02}} . \tag{2.36}$$

If one reassigns the bits such that

$$\begin{aligned} R_1 &= R_{01} + \Delta R \\ R_2 &= R_{02} - \Delta R \end{aligned} , \tag{2.37}$$

then, by a local linearization of the R(D) curves:

$$\begin{aligned} &Dtot_{new}(R_{tot}) = \\ &= \sum_{n=0}^{M-1} D_n(R_{o,n}) + \left(\left|{}^{dD_2}\!\!/_{dR}\right|_{R=R_{02}} - \left|{}^{dD_1}\!\!/_{dR}\right|_{R=R_{01}} \right) \cdot \Delta R \\ &< Dtot_{nom} \end{aligned} \tag{2.38}$$

This shows that by transferring bits from a block operating at a lower gradient, to one at a higher gradient, the global distortion is diminished, with the total rate unchanged.

Furthermore:

$$\begin{aligned} \left|{}^{dD_1}\!\!/_{dR_1}\right|_{R_1=R_{0,1}+\Delta R} &< \left|{}^{dD_1}\!\!/_{dR_1}\right|_{R_1=R_{0,1}} \\ \left|{}^{dD_2}\!\!/_{dR_2}\right|_{R_2=R_{0,2}-\Delta R} &> \left|{}^{dD_2}\!\!/_{dR_2}\right|_{R_2=R_{0,2}} \end{aligned} , \tag{2.39}$$

Thus, for a certain value of ΔR, there exists a point where

$$\left|{}^{dD_1}\!\!/_{dR_1}\right|_{R_1=R_{0,1}+\Delta R} = \left|{}^{dD_2}\!\!/_{dR_2}\right|_{R_2=R_{0,2}-\Delta R} , \tag{2.40}$$

From the above it is evident that this condition corresponds to the lowest distortion for a given rate.

This process may easily be extended to account for all N tiles, where the two tiles in question at each point are those with the highest and lowest gradient. After sufficient iterations all N tiles will have the same gradient, which it is clear, corresponds to the operating condition of minimal distortion for a given rate constraint.

This proves the hypothesis; minimising the Lagrange cost, for a given λ, over each tile, minimises the distortion of the entire frame, for a certain rate. The value of λ can be changed until the total rate over the N blocks equals the desired rate for the frame.

2.2.4 Summary of Lagrange Multiplier Technique

Section 2.2 has given a thorough treatment of the Lagrange multiplier technique, from theory to practice. This technique is very important in image and video coding as it provides a constructive framework within which RD optimisation may be performed.

2.3 Summary and Conclusion

This chapter has provided an introduction into rate distortion estimation and optimisation.

Source Modelling

The fundamental theory of RD estimation based on Shannon's formulation was outlined briefly. The problems encountered when using this theory in practice, due to the difficulty of providing statistical models of complex sources such as video, and the non-constructive nature of the theory were presented. An alternative approach, operational RD theory was presented, but the fundamental problem of computational complexity was highlighted.

Methods of RD estimation that are used in practice were then explored. The rate control methods of MPEG-2, MPEG-4 and H.263 were examined. Based only on the simplicity of the models used, it is expected that these algorithms will perform poorly. Thereafter the new method of ρ-domain analysis was presented in detail. The performance of this algorithm is compared to the MPEG algorithms and found to be greatly superior. Due to this method's strong characteristics, it will form part of the implemented algorithm, as described in Chapter 6.

RD Optimisation

The process of obtaining the minimum distortion, compressed, representation of a source, based on an RD estimation was presented. As this RD optimisation is usually carried out using the Lagrange multiplier, a full development of this theory was provided. Its application to both single, and multiple unit coding was developed.

Overall

The tantalising prospect of creating an RD optimal coder has for many years floundered on the inability of RD theory to provide an accurate and robust forward estimate of the RD performance of a generalised source and coder. Recent advances bring this goal closer by providing good RD estimation under general conditions.

The algorithm to be presented in Chapter 6 will utilise the ρ-domain RD estimation theory, and the Lagrange optimisation methods, due to their remarkable performance, as outlined in this chapter.

Chapter 3 - Wavelet Still Image Compression

This chapter aims to produce a comprehensive literature survey on the state of the art of still image compression using wavelet techniques. Each of the leading ideas in the field will be presented, and algorithms that have been built on these ideas will be compared based on published results. The goal of this chapter is to discover what factor has enabled wavelet based image coders to outperform the current algorithms.

There are three stages in the image compression process; the wavelet transform, coefficient quantization and entropy coding:

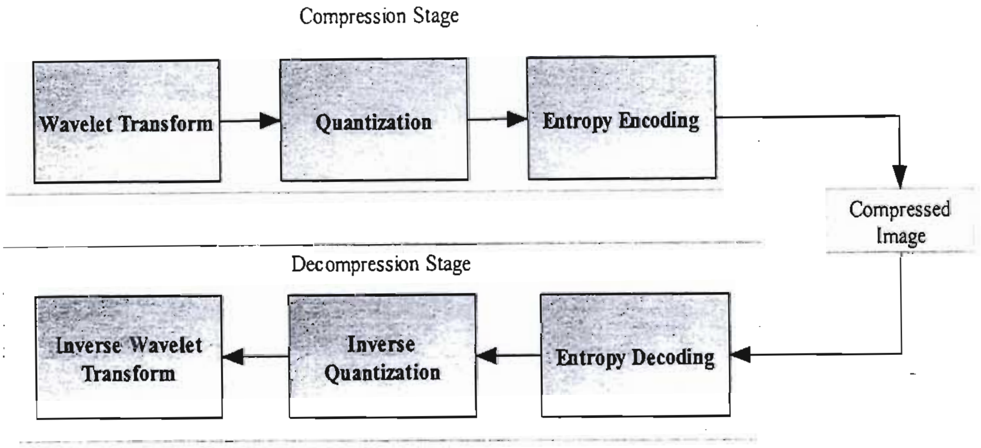


Figure 3-1: Stages of a Compression Algorithm

This chapter will explore the wavelet transform block, and the quantization and entropy coding blocks, to determine the effect on performance of each stage. This shall be done through extracting algorithms from the literature that operate in each one of these stages, and comparing their published results.

The first, wavelet transform, stage transforms the image data from the spatial domain to the wavelet domain. The major choice in this stage is the nature of the basis function, or wavelet. This is a critical component of the system in determining compression performance. A successful wavelet transform will capture the image features, providing a succinct representation; that is, a high coding gain. This stage removes linear dependencies within the data.

The second stage concerns itself with quantising the coefficients produced by the wavelet transform. This stage performs the lossy compression, thus it will yield the greatest returns on computational effort invested. Here, techniques to select the most relevant coefficients based on known statistical and other models are utilised.

The third stage of entropy encoding utilises the well known techniques of Huffman and arithmetic encoding, to losslessly encode the quantised coefficients of the previous stage. In some cases this stage is combined with the quantisation stage, such algorithms will also be explored below.

Section 3.1 details the quality metric by which the various compression methodologies will be compared. This section is merely provided to allow the various algorithms presented afterwards to be compared.

This is followed by Section 3.2 which explores the wavelet transform block. Various techniques which modify the wavelet transform itself are drawn from the literature and their effect on the compression performance is analysed.

Finally, Section 3.3 discusses the major quantization and entropy coding methods used in the wavelet domain to achieve compression. This discussion is structured according to the image model used, and in addition to reviewing the many compression algorithms available, provides discussion on the importance of accurate source modelling in compression. The early algorithms are covered for introductory purposes, and the discussion proceeds to increasingly modern methods. This section forms the bulk of this chapter, as it is these coding strategies that take advantage of the correlation structures exposed by the wavelet transform to achieve compression.

3.1 Image Quality Measurement

In order to make a meaningful comparison between compression techniques, a measure of the degradation suffered by an image through the compression process is required. This has become a notoriously nebulous problem, as the perceived quality of an image is dependant on its usage and viewer. Furthermore, the quality of an image has several components and the general problem of producing a cost function incorporating all of them is assumed intractable. Nevertheless, in order to make comparisons between compression schemes, several metrics have been proposed. In this work, Peak Signal to Noise Ratio, PSNR, which is measured in decibel, dB, will be used.

3.1.1 Peak Signal to Noise Ratio (PSNR)

PSNR indicates the total error present in a processed image. The mean square error, MSE, of an image is defined as

$$MSE = \frac{\sum_{x=1}^{XSIZE} \sum_{y=1}^{YSIZE} \left(p_{x,y} - \hat{p}_{x,y}\right)^2}{XSIZE \cdot YSIZE}. \tag{3.1}$$

Where XSIZE and YSIZE are the dimensions of the image in pixels, $p_{x,y}$ represents the pixel at location (x,y) in the original image, and $\hat{p}_{x,y}$ the same pixel in the compressed image. Thus MSE provides an aggregated error measure for an entire image.

PSNR is based on this metric, and defined as

$$PSNR := 10\log_{10}\left(\frac{255^2}{MSE}\right) dB, \tag{3.2}$$

assuming an 8-bit image, hence the normalization by 255. PSNR is a normalised quality metric, based on MSE, which is a distortion metric.

The use of MSE and PSNR is justified from the literature. For instance the JPEG 2000 project found that MSE minimizing codecs performed well perceptually, as well as numerically. Indeed it is standard practice to publish performance results in PSNR. Thus, using this measure allows this work to be contextualised within the existing literature.

3.2 Basis Functions

It has been argued [Saha99] that the success of the wavelet transform in image compression is due to the good representation afforded to image artefacts by the wavelet basis functions. [Wu01] demonstrates that the wavelet transform exposes spatial similarity structures, as will be explained in Section 3.3.2. Furthermore, certain image artefacts are best described spatially (edges for example), and others in the frequency domain (uniform textures.) The dual space-frequency (scale) representation offered by the wavelet transform thus allows efficient representation of each. This argument will be revisited in Section 3.3.

The standard wavelet representation structure uses a single mother wavelet, and a dyadic decomposition tree (to be described). However, the number and nature of mother wavelets, and the decomposition structure are subject to design. This discussion will highlight the effects of these alterations. Owing to space limitations, this discussion is unfortunately brief and incomplete. Three techniques are however described as an overview of the topic.

Section 3.2.1 describes how variability of the basis function and decomposition tree affects compression. Section 3.2.2 describes how an image may be projected onto a number of different wavelet bases simultaneously, and the utility of this representation. Finally, Section 3.2.3 describes the projection onto a non-separable, two dimensional basis function.

Appendix A provides a summary of the mathematics of the wavelet transform which is used for this discussion.

3.2.1 Best Bases

Within the widely used wavelet decomposition structure, either the mother wavelet or the decomposition tree may be altered. Both of these changes amount to changing the basis set.

As Appendix A shows, a signal $f(x)$ that undergoes wavelet decomposition is projected onto the basis set

$$\{\phi_{j,n}(x)\} \text{ and } \{\psi_{j,n}(x)\}. \tag{3.3}$$

It is obvious that modifying the mother wavelet functions $\phi(x)$ and $\psi(x)$, changes the basis set. Furthermore, the Mallat construction specifies that at each level, only the lower frequency band is decomposed further. Thus the scaling function bases $\phi_{j,n}(x)$ will not exist at all values of j. By allowing each subband to spilt, $\phi_{j,n}(x)$ will exist for these values of j, thus expanding the basis set. As will be shown below, there are other ways to modify the basis set by modifying the decomposition tree.

This section explores briefly the effect of these modifications to the basis set.

3.2.1.1 Modification to the Mother Wavelet and Scaling Function

The first tactic discussed is modifying $\phi(x)$ and $\psi(x)$. The goal is to produce a basis set that decorrelates the image data best. Stated differently, the goal is to find a mother wavelet that provides the best joint energy concentration in the spatial and frequency domains. Several works have investigated this avenue.

Notably [Saha99] presents strong empirical evidence that suggests that the choice of mother wavelet is actually not a critical factor. Several test images, identified in the columns of the table (Lena, Barbara etc) were compressed using a scheme similar to [Taub94]. In each case a different wavelet mother function (the row headings Haar to CDF-3/13) was used. The following table, drawn from their results, shows the PSNR of each image compressed at 16:1 with each mother wavelet as well as a statistical summary across the images.

Image Statistics	Lena	Barbara	Baboon	Peppers	Bengali	mri	nervecell
Mean	99	117	129	120	213	57	117
Median	97	117	130	121	255	44	98
Std. Dev.	53	55	42	54	78	54	84
Variance	2796	2982	1789	2894	6116	2900	7001
Wavelets							
Haar	31.51	27.23	24.19	33.04	27.3	29.42	30.66
Daub2 (4 coeff)	33.03	28.32	24.77	34.37	24.92	31.59	32.4
Daub4 (8 coeff)	33.48	29.1	25.04	34.74	24.25	32.31	32.73
Daub8 (16 coeff)	33.65	29.64	25.1	34.38	22.77	32.15	33.04
Adelson (9 coeff)	33.93	29.51	25	34.78	24.66	32.38	33.6
CDF-9/7	34.28	29.54	25.05	35.19	24.71	32.77	33.96
(CDF-7/9)	33.1	28.76	24.47	34.36	24.58	31.39	32.83
CDF-9/11	33.97	29.8	24.68	34.68	24.21	32.28	33.46
Odegard-9/7	34.3	30.16	24.98	35.08	24.62	32.73	33.9
Brislawn-10/10	33.68	29.07	24.25	34.53	23.07	32.63	33.71
Villasenor-10/18	34.15	30.09	25.33	35.19	23.21	32.85	33.96
(Villasenor-18/10)	33.41	29.2	24.65	34.39	24.36	31.67	32.85
Villasenor-13/11	34.28	30.18	24.86	34.99	24.47	32.67	33.66
(Villasenor-11/13)	33.82	29.39	24.75	34.57	24.6	32.11	33.49
Villasenor-6/10	34.04	29.57	24.57	35.16	23.85	32.73	33.72
(Villasenor-10/6)	33.14	28.75	24.5	34.15	24.84	31.26	32.61
CDF-13/3	33.65	28.81	24.78	34.69	24.51	32.3	33.58
(CDF-3/13)	32.98	28.54	23.82	33.85	23.75	31.04	32.24
Comp. Statistics							
Max	34.3	30.18	25.33	35.19	27.3	32.85	33.96

Min	31.51	27.23	23.82	33.04	22.77	29.42	30.66
Max Difference	2.79	2.95	1.51	2.15	4.53	3.43	3.3
CDF 9/7 Difference	0.02	0.64	0.28	0	2.59	0.08	0

Table 3.1: Perfomance Comparison between Wavelet Bases [Saha99]

For a single image the maximum difference between wavelet choices is shown, and is seen to be quite variable, suggesting that adaptivity of the mother wavelet may yield returns. However, considering the difference between the best performing wavelet, and the standard Daubechies 9/7 (CDF 9/7 in the table), which is insignificant in all but one image, this author believes that neither custom design of the basis functions, nor adaptive basis function will yield substantial returns.

3.2.1.2 Adaptivity of Subband Selection (Wavelet Packets)

In this scenario, the action taken is to select the R-D optimal subband tree expansion. Classically, the wavelet transform is performed using the dyadic (Mallat) construction. That is, the original spectrum is split in two; then the lower subband, is split into two further subbands. This process is iterated, in each case splitting only the lower subband. An alternative approach, termed packet decomposition, considers splitting both the high and low subbands at each iteration, as illustrated below.

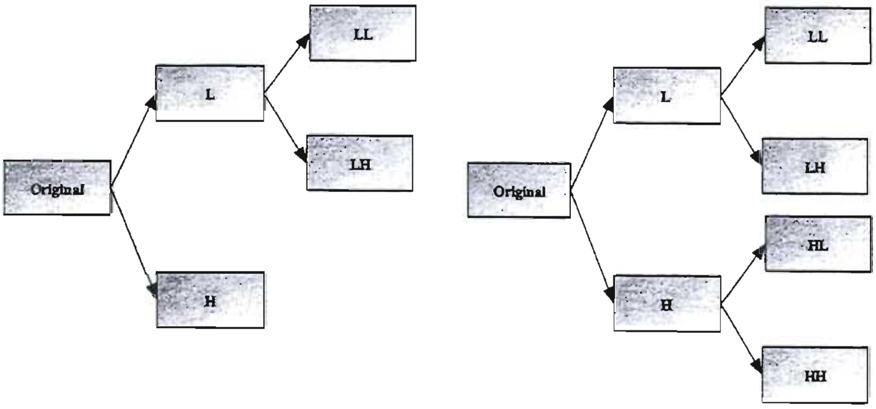


Figure 3-2 : Two Level Mallat- and Packet- Wavelet Decomposition

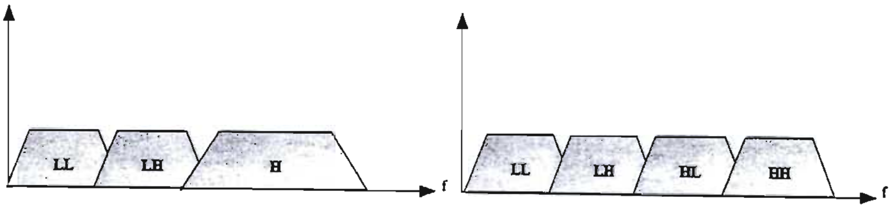


Figure 3-3: Two Level Mallat- and Packet- Decomposition Spectra

3.2.1.3 Best Wavelet Packet Bases

Ramchandran and Vetterli [Ram93], present a technique they call Best Wavelet Packet Bases. The algorithm is based on rate-distortion theory, an introduction to which is given in Chapter 2. The algorithm operates by expanding each node at each level of decomposition, generating a full packet decomposition to some level. Then the algorithm traverses the tree, removing a level of decomposition if it fails a rate-distortion test. This removal of a level of decomposition is termed pruning, and is shown in Figure 3-4.

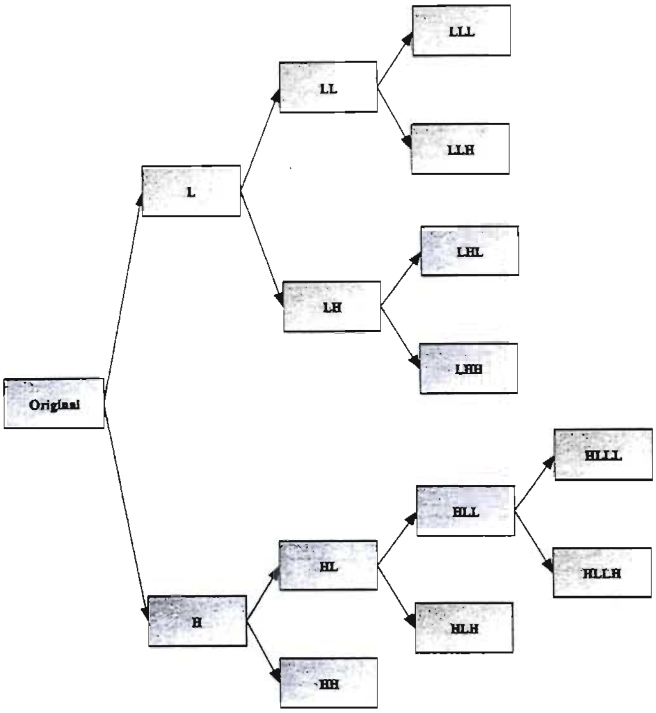


Figure 3-4: Example Pruned Wavelet Packet Representation

This pruned packet decomposition structure represents a rate-distortion optimal basis set, for a given image and mother wavelet function.

This paper reports results for the 'Barbara' image, which is characterised by more high frequency energy than usual natural images. This is owing to texture in the image. Standard image coders consequently struggle to compress it well. The paper compares the performance between standard wavelet decomposition, followed by scalar quantization system, with the pruned packet tree decomposition, followed by a scalar quantization system. The latter performs 3-4dB better than the former, at a given rate.

In addition the authors found that by segmenting the original image into tiles, and treating each independently, up to 1dB further gain may be made. This is to be expected owing to the spatially varying nature of image statistics.

3.2.2 Multiwavelets

In the schemes described previously only one wavelet and its scaling function are used at a time; these are termed scalar wavelets. Multiwavelets, however, use multiple wavelet and scaling functions in the decomposition. This allows extra degrees of freedom in the design, which allow multiwavelets to be symmetric and orthogonal, simultaneously. Scalar wavelets cannot posses both these properties concurrently.

This suggests that multiwavelets may be a more powerful image compression tool than scalar wavelets. However, at this time, multiwavelets have only produced results at best equivalent to scalar wavelets, and then at greater computational cost, [Iyer01], [Mart01].

[Iyer01] finds that balanced multiwavelets uniformly perform 0.2 – 0.7dB worse than good scalar wavelets. [Mart01] reports mixed results; with natural images the scalar wavelets performed better, while multiwavelets compressed computer generated images better. This may be on account of texturing.

[Mart01] also develops a new procedure combining multiwavelets with packet decomposition, as in the scalar wavelet case. It was found again that the multiwavelets performed better with synthetic images, and the scalar wavelets performed better with natural images.

In generating these results, SPIHT is used to quantize both the scalar and multiwavelet representations. SPIHT is designed for scalar wavelet representations, and the authors note that should a new quantization scheme be designed for the multiwavelet case, results should improve.

3.2.3 2D Bases

In processing images, it has been the traditional practice to consider the image dimensions separately. That is, to filter the image horizontally, line by line. Then, the coefficients obtained this way are filtered vertically, column by column. However, this technique is considered sub-optimal for reasons explored in [Vett01]. Given that images are two dimensional signals, projecting them onto a 2D basis function seems natural. An example of such a 2D wavelet is shown in Figure 3-5.

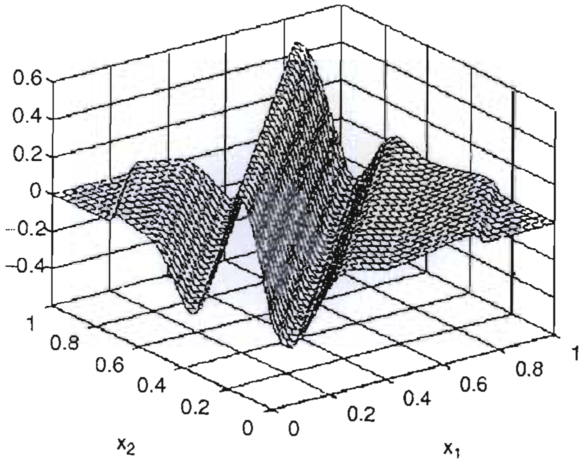


Figure 3-5: Example of a True 2D Wavelet |Vett01|

Such 2D bases include ridgelets, curvelets and bandelets. However, as noted in [Vett01], this field is very new, thus quantization methods that take advantage of this representation do not yet exist.

3.3 Still Image Quantization and Entropy Coding Methods

Section 3.2 explored the role of the wavelet transform in achieving compression performance. Figure 3-1 shows that quantisation and entropy coding follow the wavelet transform, and this section will explore these topics.

It is conceivable that the wavelet transform itself is mainly responsible for the performance of wavelet based image compression algorithms. The first task of this section is to refute this idea, by showing that the wavelet transform exposes many correlation structures, but that quantisation and entropy coding are responsible for exploiting these structures to achieve compression. Section 3.3.1 propounds this argument.

Wavelet image coding entered the realm of research in the1980s. The first successful coders were developed in the early 1990s; the seminal paper for this period of research was written by Antonini, Barlaud, Matthieu and Daubechies, [Anto92]. Ingrid Daubechies is a well known mathematician responsible for the design of several wavelets, and wrote the standard text on the field, "Ten Lectures on Wavelets," [Daub92].

These early algorithms operated mainly in the frequency(correctly, scale) domain. They are based on the traditional subband coding model of images. This model assumes that an image decomposed into frequency bands will have coefficient distributions in each band that are independent and identically distributed, according to either Laplacian or Gaussian models.

These codecs focused on producing scalar or vector quantisers optimized for each subband of the wavelet transform. Given the variance of each band, the quantization step-size or codebook is designed optimally for each subband. In this regard they are most similar to subband-coding schemes, except that the wavelet transform is used to produce the initial subbands, rather than the Fourier transform. In all of these methods, advantage is taken of the fact that most image energy is localized in the low frequencies

Although these subband-like algorithms are efficient at encoding images based on their assumed models, the model itself is in fact inadequate. [Xiong99] proposes that there are two important properties that most natural images exhibit:

1	Most of the signal energy is contained in the low frequencies.
2	Images show local statistical stationarity; the area within an object is typically self-similar and is surrounded by a discontinuity or edge. This is manifest in the wavelet domain by areas of low frequency, surrounded by spatially concentrated, high frequency, artefacts.

Table 3.2: Wavelet Codec Image Model

Subband-like algorithms account for the first property but not the second. This property is a dual space-frequency(scale) characteristic, and thus the wavelet transform is ideally suited to its capture.

Sections 3.3.2 to 3.3.11 will explore recent image coding techniques that explore both properties of images. The mechanism that first rose to prominence is termed zerotree quantization, which is introduced by Shapiro [Shap93] in his paper describing his image codec, EZW (Section 3.3.2). The zerotree quantization method has been almost universally employed in recent wavelet codecs, owing to its high performance.

More recently the statistical model underlying EZW has been found limiting. Thus, higher order statistical models than that employed by EZW are being deployed; schemes in this genre are explored in Sections 3.3.8 to 3.3.11.

The sections that follow will examine the prominent codecs and their operational principles. Coding results will also be given, in PSNR, so that an objective comparison between the various methods may be made.

Section 3.3 may be organised according to the following chart:

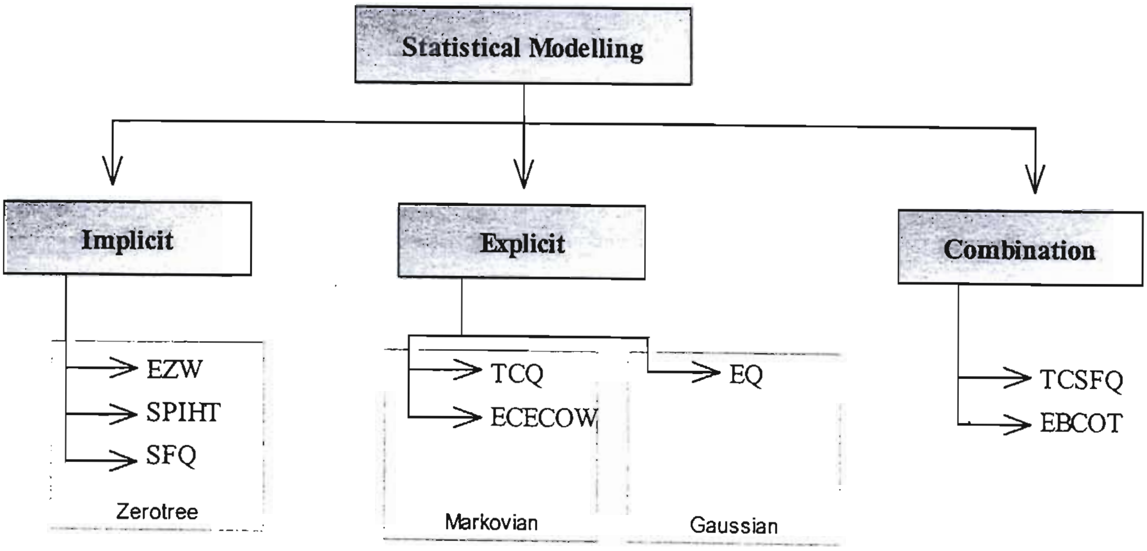


Figure 3-6: Method Taxonomy

All the methods presented are quantisation modes, except for ECECOW and EQ that are both entropy coding methods. This discussion will start with the first wavelet coding method, EZW, and progress from left to right according to the chart above, which approximates increasing sophistication.

The underlying theme is that all these methods rely on an identical wavelet transform, but through recognising and exploiting different correlation structures different compression performance is achieved. This highlights the argument that proper source modelling a most crucial aspect of any compression system.

3.3.1 Utility of Source Modelling

The argument has long been that wavelets outperform the DCT due to the superiority of the basis set[Saha99]. Indeed, Section 3.2 has this as an implicit argument. However, Xiong, Ramchandran, Orchard, and Zhang [Xiong99], present evidence that this is not necessarily the case.

They argue that all comparisons to date have been made between wavelets and baseline JPEG. Thus both the transform and quantisation stages are different. Inferring results about only one stage is therefore logically unfounded. The authors present an experiment in which the quantisation stage of the JPEG codec is replaced with a wavelet like quantiser based on SPIHT (Section 3.3.3). Then the standard 'Lena' and 'Barbara' test images are compressed using baseline JPEG, the new DCT algorithm, and the representative wavelet algorithm, SPIHT. The following results were produced:

	PSNR (dB)					
	JPEG		New DCT		Wavelet	
Rate (bpp)	Lena	Barbara	Lena	Barbara	Lena	Barbara
0.25	31.6	25.2	32.3	26.7	34.11	27.58
0.5	34.9	28.3	35.9	30.6	37.21	31.39
1	37.9	31	39.6	35.9	40.4	36.41

Table 3.3: Transform Stage Performance Comparison[Xiong99]

The results show the dramatic improvement in the DCT codec through the use of a recent quantiser. Indeed the performance gain brings the DCT almost on par with the wavelet scheme. The conclusion drawn, is that it is the quantization of the coefficients that provides the performance gains of recent codecs, not the choice of basis function.

This assertion is borne out in by the codecs that follow. The wavelet domain exposes image characteristics that allow sophisticated statistical context modelling and quantization. Wu [Wu01] explains that while the transform stage may remove linear dependencies between samples, higher order modelling allows more complex dependencies to be exposed and exploited.

3.3.2 Embedded Zerotree Wavelet Coding (EZW)

This ground breaking algorithm, introduced by Shapiro in [Shap93], is the first of the recent wavelet codecs, and hence taken as our point of departure. It efficiently captures both the energy concentration in the low frequencies and high frequency energy clustering. This is achieved by the method of zerotree quantization.

3.3.2.1 Premise

Each subband in the 2D wavelet transform contains different frequency information, but the same spatial information, barring resolution differences. Thus, the same region in space is represented in each subband. The inter-band structure representing one region in space looks like a tree, as illustrated below.

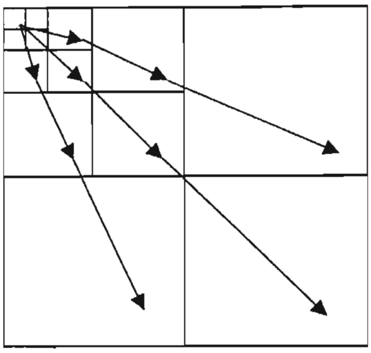


Figure 3-7: Wavelet Domain, Spatial Similarity Tree

Furthermore these trees exhibit strong statistical properties. Shapiro shows that although the wavelet transform removes the linear correlations between coefficients as expected, there is still a strong correlation between the square of the coefficients; this is termed spatial self-similarity. In fact, he shows that under mild restrictions it is statistically probable that given that a parent node falls below a certain threshold, all of its descendants do as well. This is the basis for the EZW approach; given a threshold T, it is statistically feasible to predict the behaviour of coefficients that are less than T. Based on this, a new form of quantisation; zero-tree quantisation, is introduced. A zero-tree is a tree-structure, such as in Figure 3-7, in which all the coefficients are below a given threshold, hence considered zero.

The wavelet subband decomposition separates the low frequencies for encoding, thus allowing the first property of the image model in Table 3.4 to be exploited. Edges produce high value coefficients in the high frequency (or low scale) subbands. The zerotree quantisation method aids their description by quantizing non-edge features to zero very succinctly, thus exposing these edges. This accounts for the second property Table 3.2.

3.3.2.2 Algorithm

As the wavelet transform is unitary[Said96], the largest coefficients in the wavelet domain are the most important; they represent bases with more signal energy. Thus, in compression it is desirable to transmit these coefficients first, as they will produce the greatest decrease in distortion of the reconstructed image. Thus a threshold is defined, initially equal to the largest coefficient in the image.

The algorithm operates by scanning through each subband, starting at the lowest and progressing to higher subbands in a raster scan order, as illustrated below:

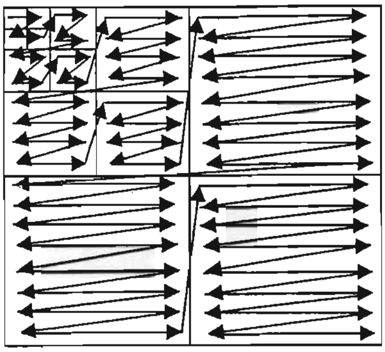


Figure 3-8: EZW Raster Scan Order

During the raster scan, at each coefficient, a symbol is output indicating whether the coefficient is greater than the threshold. In addition, each element of the tree descending from the coefficient is compared to the threshold, and should they all be insignificant, a zerotree symbol is output. The raster scan then continues through the image, excluding all coefficients previously found to be elements of a zerotree. This allows the output symbol stream to be shorter.

After the scan has been completed, this set of symbols is either Huffman or arithmetic encoded, which efficiently encodes the spatial localization of high frequencies at edges. This is owing to long sequences of zerotree symbols in the non-edge areas, which may be coded efficiently with entropy coding.

In addition a separate stream is generated, where for each significant coefficient, the difference between this coefficient and the threshold is transmitted.

Once a full scan has been completed, the threshold is lowered, and the process repeated, excluding those coefficients already found to be significant. Through this iteration, eventually all the coefficients will be transmitted.

A crucial element of the system is termed the bit plane transmission. Rather than transmitting the numerical difference between a significant pixel and the threshold, the bit from the same bit plane as the threshold (ie the bit indicating 256, 128, 64 etc) is transmitted, for each coefficient found significant. This is termed order bitplane transmission and is most effective.

3.3.2.3 Results

The results shown here, as in other tabulations are drawn from the UCLA Image Communication Laboratories Results Page [UCLA1] and the original work; all images used are 512 x 512 greyscale images.

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	33.17
Lena	0.5	36.28
Lena	1	39.55
Barbara	0.25	26.77
Barbara	0.5	30.53

Table 3.4 : UCLA PSNR Results for EZW[UCLA1]

3.3.2.4 Discussion

The algorithm presented is efficient at encoding both elements in the image model presented in Table 3.2.

The dominance of low frequencies in images is accounted for by the raster scan order, which encodes low frequencies first, as they are most likely to be significant.

The high frequencies representing edges are also accounted for. Edges will produce large coefficients in the high frequency subbands, but not necessarily in the low frequency bands. By searching the tree structure descending from each coefficient, these edge coefficients are found. The zerotree quantization method efficiently removes large tracts of insignificant coefficients, without removing significant edge coefficients.

Although the results presented above are the worst of all recent wavelet coders, this is because this is the original work, upon which all improvements are based. This image codec is of great importance as it introduced the zerotree quantization method that successfully extends the subband coding functionality of capturing energy localization in low frequencies, to the capture of high frequencies which are spatially localized on edges. The utilization of zerotrees is improved in later algorithms.

3.3.3 Set Partitioning in Hierarchical Trees (SPIHT)

The most celebrated codec is Said and Pearlman's SPIHT presented in [Said96]. It is based heavily on the EZW algorithm of Shapiro, [Shap93], but features more sophisticated stream generation. This algorithm has been the benchmark for comparison of all later codecs, and the basis of our own, thus details of its operation will be given.

3.3.3.1 Premise

While SPIHT uses the same image model as EZW, the algorithm is more efficient in constructing the output bitstream and thus yields higher performance. Said and Pearlman note that minimising the distortion in the transform domain will minimise the distortion in the original domain too. Thus, the major change in SPIHT over EZW is to alter the order in which the algorithm scans to ensure that the largest coefficients are transmitted first. While EZW follows a raster scan order, SPIHT includes various tree searching routines.

3.3.3.2 Algorithm

SPIHT operates on three sets:

List of Significant Pixels (LSP)	All pixels whose value is above the threshold.
List of Insignificant Pixels (LIP)	All pixels that are part of a significant tree but are themselves below the threshold.
List of Insignificant Sets (LIS)	All pixels that are the root of zerotrees.

Table 3.5 : Sets in the SPIHT algorithm

Implied in this is a fourth set, consisting of all those pixels that are below the threshold, and lie in a zerotree. However, these are never explicitly addressed.

The algorithm begins by searching the image for the largest element, which is termed the threshold. Any coefficient less than the threshold is termed insignificant, and any greater than or equal to the threshold is termed significant. It then searches the lowest subband for all the significant coefficients; and adds them to the LSP and outputs their sign; the rest are added to the LIP. In addition, the tree stemming from each of these coefficients is searched for significance. Those pixels, whose entire descendant tree is insignificant, are added to the LIS.

The algorithm then scans through the LIP, outputting the significance of each element. In successive scans, as elements become significant they are moved to the LSP, and their sign is output. The set of bits output to indicate sign information is termed the sign bit set.

Thereafter the algorithm scans the LIS, for each coefficient determining the significance of the elements of the descendant tree and outputting a bit to indicate the result. If there is any element that is significant, the next level of descendants is checked for significance and a bit output to indicate each result. The elements are also added to the end of the LSP, LIS and LIP as appropriate. The scan through the LIS will continue, and any elements that have been added to the list will be checked further. This traversal of the LIP and LIS is termed the sorting pass.

After the LIS has been fully traversed, the algorithm steps through the LSP outputting the appropriate bitplane value for each coefficient, as explained in the EZW algorithm. These bits are termed the refinement bits. The threshold is then halved, and the process repeated from the search through the LIS. The traversal of the LSP and outputting of refinement bits is termed the refinement pass.

By beginning the search in the lowest subband, the energy localization in low frequencies is exploited. The spatial localization of high frequency coefficients of significance is also exploited through the tree structure, which will quantise all those pixels in interior spatial regions, thus exposing the edges. Thus, the algorithm captures the low frequency coefficients where most energy is frequency localized, as well as the edges, where energy is spatially localized. This mechanism is identical to EZW.

3.3.3.3 Results

UCLA [UCLA1] compiled the following results from this algorithm.

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	34.11
Lena	0.5	37.21
Lena	1	40.46
Barbara	0.25	27.79
Barbara	0.5	31.72

Table 3.6 : UCLA PSNR Results for SPIHT[UCLA1]

3.3.3.4 Discussion

Although this algorithm is identical to EZW in its zerotree quantization, it achieves higher compression due to its zerotree searching mechanism. EZW raster scans through the subbands starting from the lowest, as each coefficient significance is determined and the tree expanded if it contains significant descendants. These coefficients are then only added to the outgoing bitstream when the raster scan reaches them. SPIHT scans through the lists, rather than in a raster order, and thus encodes significant descendants immediately, which yields rate savings.

Furthermore the algorithm outputs its state at each significance test; these are termed the significance bits. This results in an output bitstream that is embedded; it can be truncated at any point and reconstructed correctly to that point. This is very useful in generating specific rate streams, as Chapter 6 will demonstrate.

This algorithm in fact represents a more compact transmission strategy for the same information as EZW, which is the main reason for its improved performance.

3.3.4 Rate Distortion Optimisation to SPIHT

Pearlman's original technique guarantees lossless reconstruction at a bitplane level if all bits in that plane are transmitted. However, should the stream be terminated before the entire bitplane has been transmitted, the distortion is suboptimal. As mentioned in section 3.3.3.1, SPIHT aims to minimise the distortion, however no regard is given to the effect this has on the rate behaviour. That is, for a given number of bits the distortion is not as low as it could be. Lin and Gray [Lin02], present a technique that minimizes the Lagrangian cost function, $J = D + \lambda R$, when selecting bits from the wavelet tree to include in the transmitted stream.

As indicated in Section 3.3.3 on the SPIHT algorithm, there are three types of bits within the output stream; sign bits, refinement bits and significance bits. Sign bits are output when a coefficient is added to the LSP to indicate the coefficient sign. Refinement bits are added to the output stream during the refinement pass. Finally significance bits are the bits output to the stream indicating the result of each significance test.

Lin and Gray note that while sign and refinement bits directly contribute to a distortion reduction in the reconstructed image, significance bits do not. Thus, during the sorting pass, it will be beneficial to expand elements of the LIS which have the greatest number of significant descendants, before expanding those with fewer significant descendants. Thus Lin and Gray's algorithm performs a Lagrangian test, rather than a significance test, to the descendant tree of each coefficient in the LIS, to determine whether to expand it or not, thus maximising the distortion decrease for a given bitrate increase.

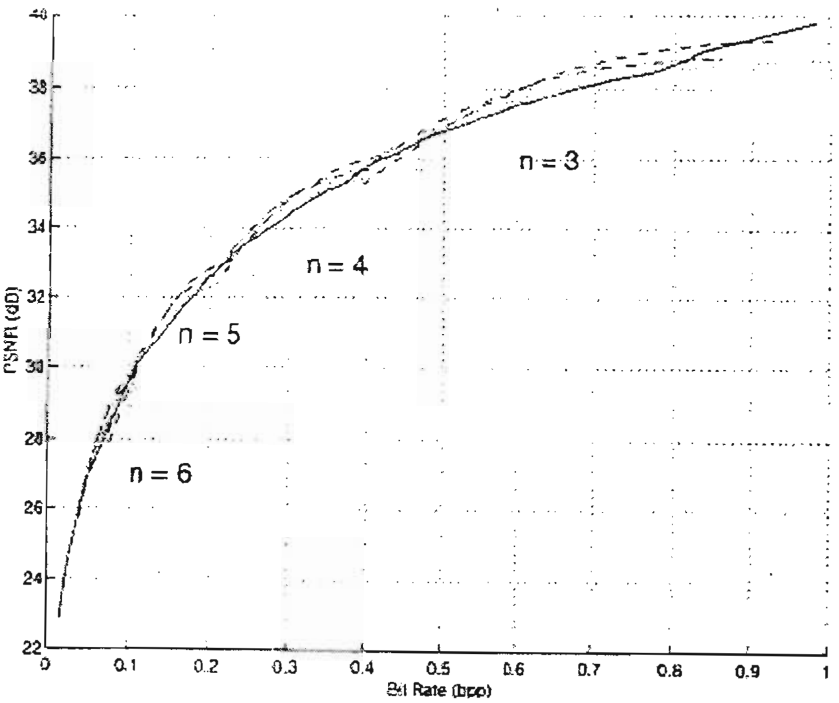


Figure 3-9: Comparison Between RD-Optimised SPIHT and SPIHT[Lin02]

Although the principle of this algorithm is promising, another similar work, SFQ, performs better and will be described in greater detail below.

3.3.5 Space Frequency Quantization (SFQ)

Continuing the theme of rate-distortion optimising approaches, Xiong, Ramchandran and Orchard in [Xiong96], present an algorithm that is based on a joint zerotree and scalar quantization optimization. Notably, due to the scalar quantization, the bitstream generated is not embedded, as in SPIHT.

3.3.5.1 Premise

Noting, as in [Lin02], that SPIHT minimises the distortion, rather than optimises the R-D behaviour, the authors design their algorithm to provide optimal R-D behaviour.

Conceptually, should a tree structure contain only insignificant coefficients except for a single significant high frequency coefficient, the tree cannot be zerotree quantized, using the EZW/SPIHT scheme. However, the distortion benefit of not zerotree coding the coefficient may be outweighed by the rate overhead involved.

Stated precisely, the SPIHT quantization algorithm does not operate at rate-distortion optimality at each quantization point. Xiong et al, redefine zerotree quantization, such that a tree is zerotree quantized according to a rate-distortion criterion, rather than a significance criterion. They then jointly optimise the zerotree criterion with the scalar quantiser stepsize, to ensure that the algorithm maintains rate-distortion optimality at each operational point.

3.3.5.2 Algorithm

The joint RD optimization may be written as:

$$\min_{\{q\in Q,S\prec T\}} D(q,S) \text{ subject to } R(q,S)\le R_b \; , \qquad (3.4)$$

where, using Xiong's notation, Q represents the set of possible scalar quantiser choices; q, a particular quantisation step-size; T, the full tree expansion of the wavelet representation and S, a pruned subset of this tree. D(q,S) represents distortion (in a MSE sense), and R(q,S) is the rate, both at the operational point (q,S). Using the Lagrange Multiplier method, as in CHAPTER 2, this equation may be rewritten, for R(q,S) = R_b:

$$\min_{\{q\in Q,S\prec T\}} \left[J(q,S) = D(q,S) + \lambda\cdot R(q,S)\right]. \qquad (3.5)$$

As q and S are highly inter-dependant; q upon S, an iterative approach must be taken. First the optimal S for a given q must be found for each q. Then, the optimal q for each λ must be found. The optimal λ is then found using a bisection search until R(q,S) = R_b. This is written:

$$\min_{\lambda\ge 0}\min_{q\in Q}\min_{S\prec T}\left[D(q,S) + \lambda\cdot(R(q,S) - R_b)\right]. \qquad (3.6)$$

This process may be seen as a joint optimization of the zerotree- and scalar- quantization modes using operational RD estimation followed by Lagrange optimisation.

3.3.5.3 Results

This algorithm performs favourably compared to those discussed to this point. The UCLA results are as follows:

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	34.33
Lena	0.5	37.36
Lena	1	40.52
Barbara	0.25	28.29

Barbara	0.5	32.15
Barbara	1	37.03
GoldHill	0.25	30.71
GoldHill	0.5	33.37
GoldHill	1	36.7

Table 3.7: PSNR Results for SFQ [UCLA1]

This algorithm is seen to outperform SPIHT, as expected.

However, the cost in computation is very large. In their paper, Xiong et al report that a simplified SFQ encoder algorithm is 400% slower than SPIHT in encoding Lena, on their computing platform. This is due to the iterative searching scheme, which is required for the optimization.

3.3.5.4 Discussion

This algorithm draws its success mainly through improving the criterion for zerotree quantization. In addition, by jointly optimising this criterion with the scalar quantiser stepsize, further gains are made.

It has been noted that the advantage recent wavelet coders gain over subband coders, is due to their superior representation of high frequency localization in space through the zerotree mechanism. This algorithm is thus a suitable direction of enquiry as it seeks to build on the main strength of the wavelet scheme, by identifying regions of clustered high frequency artefacts preferentially to isolated coefficients.

This algorithm provides a conceptually simple, but computationally expensive approach that produces good results.

3.3.6 Stack Run Coding (SRC)

Initially proposed in 1996 by Tsai et al, [Tsai96], Stack Run Coding is a low complexity wavelet transform coefficient compression scheme. Unlike previous methods it does not rely on the zerotree quantisation paradigm.

3.3.6.1 Premise

Tsai argues that in the case of high ratio compression, most of the coefficients will be quantized to zero, particularly in a bitplane traversing method. Thus the overhead of identifying zerotrees may be unjustified. Instead the SRC coding algorithm relies on runlength coding of zeroes.

In addition, an early adaptive arithmetic coder is presented, which considers the magnitude and runlength datastreams separately.

3.3.6.2 Algorithm

The algorithm is very simple. Each significant coefficient of an already quantized wavelet representation is replaced by a structure (a,b), where (a) represents the runlength of zeroes before the coefficient, and (b) represents the signed value of the significant coefficient. The set {a} and set {b} are then independently arithmetically encoded, to take advantage of the different statistical nature of each set.

3.3.6.3 Results

Results for this method are good, given the simplicity of the algorithm:

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	33.63
Lena	0.5	36.79
Barbara	0.25	27.39
Barbara	0.5	30.98

Table 3.8 : PSNR Results for SRC[Tsai96]

3.3.6.4 Discussion

The method is outperformed by all others except EZW. However, that it outperforms the significantly more sophisticated JPEG algorithm lends credence to the move towards wavelet image compression.

An unexpected twist is the computational overhead of the algorithm, which was published as being 13% greater than the SPIHT algorithm. As this figure is based on running time of an implementation, one can site possible implementation issues as the cause.

3.3.7 Trellis Coded Quantization (TCQ)

Trellis Coded Quantization borrows from Trellis Coded Modulation, which is used extensively in channel coding. TCQ was proposed,[Marc90], before zerotree quantization, and is used in subband image coding to quantize the output bits from the wavelet transform stage.

3.3.7.1 Premise

3.3.7.1.1 Gauss Markov Processes

A Markov process is defined, [Papo91] as one where a sample at any time is statistically dependant only on the previous sample:

$$P\{\underline{x}_n \leq x_n \mid \underline{x}_{n-1},\cdots,\underline{x}_0\} = P\{\underline{x}_n \leq x_n \mid \underline{x}_{n-1}\}. \tag{3.7}$$

A Gauss-Markov process is defined as a Markov process where each sample is identically distributed according to a Gaussian distribution:

$$f_X(\underline{x}_n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(\underline{x}_n - m_X)^2} . \tag{3.8}$$

Thus, given the sample $\underline{x}_{n-1}$, one may form an expectation of the sample $\underline{x}_n$. Based on this expectation, a codebook of possible quantization levels may be designed. Rather than design a separate codebook for every possible value of $\underline{x}_{n-1}$, the set of possible values of $\underline{x}_{n-1}$ is divided into groups, and a codebook designed for each group.

3.3.7.1.2 Set Partitioning

Assume the compression demands on a set of sample coefficients$\underline{x}$, demand that the quantiser operate at n bits per sample, that is, the quantiser will have 2^n quantization levels. Should the quantisation codebook be designed according to probability of the entire set, the quantisation levels may be rather coarse. If the input set is partitioned into 2^k groups, as justified by the Gauss-Markov assumption, a quantiser operating at (n-k) bits per sample may be designed. The system as a whole would still quantize at n bits per sample, where the output is composed of k bits to indicate the group and n-k bits to indicate the quantization level within that group.

3.3.7.1.3 Trellises

Trellis coded quantisation combines the ideas of set-partitioning with the Gauss-Markov condition to produce a very efficient quantisation scheme.

The following is the diagrammatic representation of a trellis:

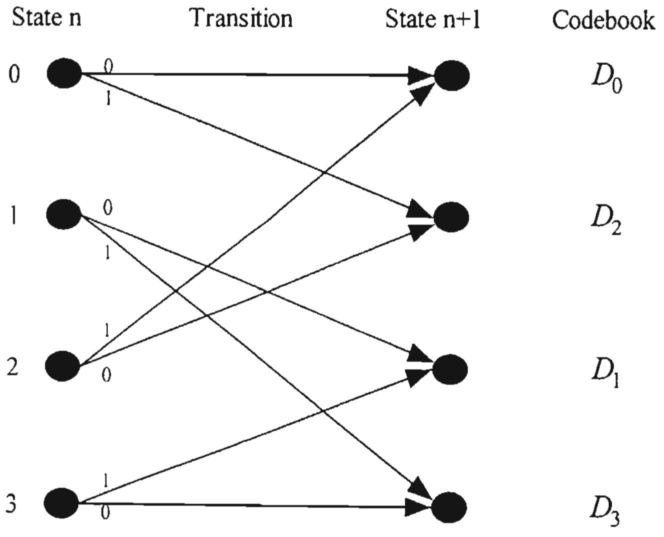


Figure 3-10: Four State Trellis [Marc90]

From one state, the system may only transition to a choice of two other states. For instance if at sample $\underline{x}_n$ the trellis is in state 0, then at $\underline{x}_{n+1}$ the trellis may only be in state 0 or state 1. The choice of states is indicated by the transition bit. Each state corresponds to the codebook used to quantise the current coefficient; in the given example $\underline{x}_n$ will be quantised with the D_0 codebook.

The utility of the trellis draws from the Markovian assumption. Given sample $\underline{x}_n$, the statistics of sample $\underline{x}_{n+1}$ are known, thus a representative codebook, D_0, should be used to quantise it. If however, the sample does not fit the expected statistics, on the next transition the trellis will transit to state 1, using a different codebook. In this way the system will traverse the trellis, using an expected representative codebook at each state. Algorithms such as the Viterbi algorithm, [Forn84], exist that traverse the trellis in an optimal fashion, these will not be explored here.

3.3.7.2 Algorithms

As TCQ was originally proposed in 1990 in [Marc90] there have subsequently been several systems based on this principle. One of the most successful was proposed by Joshi, Crump and Fischer in [Josh95]. The scheme consists of a subband decomposition (using the DCT), followed by trellis coded quantisation utilising uniform codebooks, and finally arithmetic coding. Interestingly, the scheme performs similarly to EZW [Shap93].

A more recent algorithm, also based on subband decomposition, [Kasn99], presents performance similar to SPIHT.

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	33.7
Lena	0.5	37.12
Lena	1	40.49

Table 3.9: Performance of UTQ |Kasn99|

In Section 3.3.10, a recent algorithm, TCSFQ, that utilises TCQ in conjunction with zerotree quantisation, will be discussed.

3.3.7.3 Discussion

Trellis coded quantisation provides a useful quantisation strategy for Markovian sources. This technique allows the total number of quantisation levels used to compress a source to be greater than 2^n where n is the bits per pixel, which is the limit for previous quantisation techniques. This expanded codebook allows finer quantisation, and hence superior RD performance.

However the coding gains achievable through using only this technique are not as large as with zerotree coding. However, as will be shown is 3.3.10, when TCQ is employed in conjunction with other methods, efficient coding results.

3.3.8 Embedded Conditional Entropy Coding of Wavelet Coefficients (ECECOW)

Wu presents [Wu97],[Wu01] a new method of wavelet image compression using an adaptive binary arithmetic coder based on the well-known QM coder [Penn88] that is used in JPEG1 [Wall92]. The adaptivity is driven by a high-order context modeller, which is the main contribution of the work.

3.3.8.1 Premise

Although the transform stage in an image compression system removes linear dependencies between the coefficients, higher order dependencies may still remain. The advantage of the wavelet transform is that its dual space-frequency representation exposes several dependencies through concentrating energy both in space and frequency.

Zerotree quantisation methods exploit the clustering of insignificant coefficients in well defined inter-band structures. However, other dependencies also exists, which zerotree quantisation obscures. It is the goal of the ECECOW method to adaptively capture these spatially varying structures.

Thus by building a good estimate of a coefficient based on these structures, or conditioning states:

$$\hat{P}(x_i \mid x^{i-1}), \tag{3.9}$$

an adaptive binary arithmetic coder may be driven. Using such a coder, the output code length will approach the estimated entropy:

$$-\log_2 \prod_{i=1}^{n} \hat{P}\left(x_i \mid x^{i-1}\right). \tag{3.10}$$

3.3.8.2 Algorithm

3.3.8.2.1 Context Formation

The ECECOW algorithm builds a modelling context for estimating a coefficient, based on neighbours both spatially and in the frequency domain, as illustrated below.

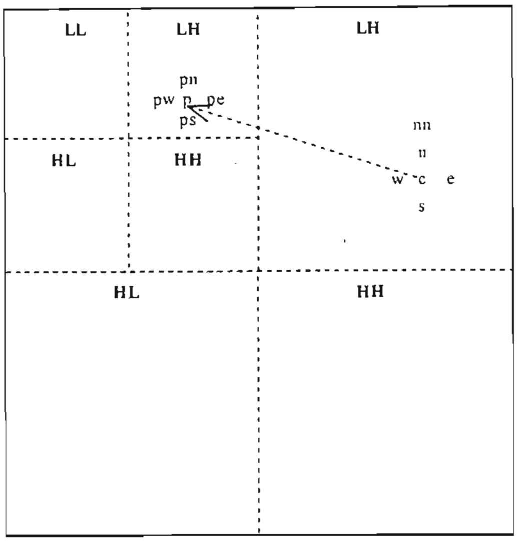


Figure 3-11: ECECOW Context Formation

The coefficient under consideration is c, and its spatial neighbours are designated by compass direction. The parent coefficient is p, with its spatial neighbours also designated by direction.

ECECOW operates on an embedded bit stream as per [Shap93] and [Said96]. Using Wu's notation the bits of a coefficient, from the MSB down to bit b, is written as $C_{..b}$.

Coefficient structures vary between subbands. For instance, in LH type subbands, vertical edges are prominent due to the wavelet filter in the horizontal direction being a high pass type. See Appendix A. Similarly HL type subbands display horizontal features. Thus the modelling context is varied per subband type. Wu finds empirically that the following contexts are effective:

$$S_{LH} = \{N_{..b}, W_{..b}, NW_{..b}, NE_{..b}, NN_{..b}, S_{..b}, P_{..b}, PN_{..b}, PS_{..b}\} \tag{3.11}$$

$$S_{HL} = \{N_{..b}, W_{..b}, NW_{..b}, NE_{..b}, WW_{..b}, E_{..b}, P_{..b}, PW_{..b}, PE_{..b}\} \tag{3.12}$$

$$S_{HH} = \{N_{..b}, W_{..b}, NW_{..b}, NE_{..b}, S_{..b+1}, E_{..b+1}, P_{..b}, C_{HL}, C_{LH}\}. \tag{3.13}$$

The extension of the context in the directions of significance is evident.

When generating the bth level of the embedded bitstream, and considering a coefficient C, only the bits in the same or previous level are considered. This is done because, as the number of possible conditioning states increases, the computational overhead of capturing statistically significant behaviour increases exponentially. This is known as the context dilution problem. Thus using only a single bit, rather than the full coefficient value, reduces the number of possible states.

3.3.8.2.2 Context Quantisation

To express the relationship between a coefficient C, and is modelling context, Wu proposes a linear estimation which he terms the local energy level:

$$\Delta_\theta = \sum_{z_i \in S_\theta} \alpha_{\theta,i} \cdot z_i, \qquad \theta \in \{LH, HL, HH\} \tag{3.14}$$

where z_i are the coefficients in the modelling context. The parameters $\alpha_{\theta,i}$ are determined by an offline regression.

The local energy level, Δ_θ provides a quantisable estimate of the coefficient C, based on the modelling context. Thus the entropy

$$H(C \mid Q(\Delta_\theta)) \tag{3.15}$$

may be minimised through careful design and optimisation of the quantiser, Q(). This process is expounded in [Wu01].

In addition to the aggregated measure (3.14), the relative significance of various neighbours was found to be of importance, thus the following measure is added to the context:

$$\underline{T_b} = [t_0, \cdots, t_4]^T = \begin{bmatrix} N_{..b} > C_{..b+1} ?0:1; \\ W_{..b} > C_{..b+1} ?0:1; \\ S_{..b+1} > C_{..b+1} ?0:1; \\ E_{..b+1} > C_{..b+1} ?0:1; \\ \begin{cases} P_{..b} + PN_{..b} + PS_{..b} > 6C_{..b+1} ?0:1, & \theta = \mathrm{LH} \\ P_{..b} + PW_{..b} + PE_{..b} > 6C_{..b+1} ?0:1, & \theta = \mathrm{HL} \end{cases} \end{bmatrix} \tag{3.16}$$

Thus, the final probability estimate that is input to the arithmetic encoder is:

$$\hat{P}\left(C_b \mid Q(\Delta_\theta), \underline{T_b}\right). \tag{3.17}$$

A similar, yet separate context formation and coding strategy is applied to the sign information.

3.3.8.3 Results

The results published from [Wu97] and [Wu01] are among the very best in the literature.

Image	Bit Depth (bpp)	PSNR (dB)	
		[Wu97]	[Wu01]

Lena	0.25	34.81	34.89
Lena	0.50	37.92	38.02
Lena	1.00	40.85	41.01
Barbara	0.25	28.85	29.21
Barbara	0.50	32.69	33.06
Barbara	1.00	37.65	38.05

Table 3.10: ECECOW Results [Wu97],[Wu01]

The major difference between [Wu97] and [Wu01] is the context quantization optimisation procedure.

In addition, various fast implementation techniques described in the papers allow the algorithm to operate with low computational overhead. The author reports a 20% speed improvement over SPIHT, while at the same time generating superior RD performance.

3.3.8.4 Discussion

This algorithm is extremely powerful. The main feature is adaptive context formation, by subband and coefficient. The algorithm exposes more complex relationships than the zerotree model, thus allowing more efficient coding through adaptive entropy coding.

By using embedded bitplane traversal, the context dilution problem is tackled, as well as allowing the very successful QM adaptive binary arithmetic coder to be employed. In addition, the computational overhead of the algorithm is low compared with other methods.

3.3.9 Estimation Quantisation (EQ)

3.3.9.1 Premise

A similar procedure is proposed by LoPresto, Ramchandran and Orchard in [Lopr97]. Based on the subband image processing literature, the authors model the wavelet coefficients as being distributed according to a spatially varying Generalised Gaussian Distribution. They estimate this distribution, and quantize and entropy encode the coefficients optimally, based on this estimated distribution.

3.3.9.2 Algorithm

3.3.9.2.1 Estimation

The method models each subband in the wavelet decomposition as a collection of independent zero-mean generalised Gaussian distribution (GGD) fields. These fields are stationary, but of unknown shape, and exhibit a spatially varying variance. The GGD is given by:

$$p(x) = \left[\frac{\nu\eta(\nu,\sigma)}{2\Gamma(1/\nu)} \right] e^{\left(-\left[\eta(\nu,\sigma)|x|^{\nu}\right]\right)} \tag{3.18}$$

where the shape parameter is given by:

$$\eta(\nu,\sigma)=\sigma^{-1}\left[\frac{\Gamma(3/\nu)}{\Gamma(1/\nu)}\right]^{1/2}. \tag{3.19}$$

The authors select a modelling context, S, that is a 3x3 or 5x5 spatial block (the size is specified by N), with the coefficient under consideration, in the centre. They state that this context may be extended to other subbands via the parent-child relationships, in a similar fashion to [Wu97].

In general, the wavelet coefficients, w_i output by the transform will be quantized, usually to bring them to integral values, $\hat{w}_i$. Given a knowledge of this quantisation method, the unquantised coefficient is known to lie between the two bin values, l_i and r_i. Hence, the maximum likelihood estimate of the sample standard deviation may be found:

$$\begin{aligned}\hat{\sigma}_s &= \arg\max_{\sigma}\prod_{i=1}^{N}P(l_i\le x\le r_i)\\ &= \arg\max_{\sigma}\sum_{i=1}^{N}\log\int_{l_i}^{r_i}\left[\frac{\nu\eta(\nu,\sigma)}{2\Gamma(1/\nu)}\right]e^{\left(-\left[\eta(\nu,\sigma)|x|^{\nu}\right]\right)}dx\end{aligned} \tag{3.20}$$

This is solved using an iterative numerical method, see [Lopr97] for details.

3.3.9.2.2 *Quantisation*

Based on the estimation of the standard deviation presented above, and the predetermined shape parameter for the subband of interest, the coefficient is quantised using an optimal quantiser.

The solution presented is to generate a sufficiently large number of quantization tables, for unity variance GGDs of different shape parameter, offline. These tables are stored as lookup tables in the encoder and decoder. The tables are then scaled online by the estimated variance.

3.3.9.3 Results

Results obtained using this method are impressive.

Image	Rate(bpp)	PSNR(dB)
Lena	0.25	34.57
Lena	0.50	37.69
Lena	1.00	40.89
Goldhill	0.25	30.76
Goldhill	0.50	33.42
Goldhill	1.00	36.96

Table 3.11: EQ Performance Results [Lopr97]

Numerical computational performance results are not given, however the authors claim the method to be 'significantly' faster than SFQ. That SFQ is a computationally complex algorithm detracts from this measure.

3.3.9.4 Discussion

This method operates within the subband image processing school. It draws advantage from adaptive context estimation with joint quantiser optimisation. By performing the quantiser optimisation offline, significant computational gains are made.

3.3.10 Trellis Coded Space Frequency Quantisation (TCSFQ)

Figure 3-1 shows the three stages in a typical wavelet compression algorithm. TCSFQ [Xiong99,2] combines the most successful method applied in each one of these stages to produce a complex, but high performance codec.

3.3.10.1 Premise

The three stages of compression have been considered separately in previous algorithms. In this method the final two are considered jointly. In addition, as many compression methods that operate in each stage utilise a different redundancy in the data, they may be applied in conjunction with one another to yield superior performance. [Xiong99,2] presents the contribution of each improvement to the final codec performance.

Each of the stages and respective methods will be considered below. In order to ease the discussion, each block will be independently covered.

3.3.10.2 Wavelet Transform

As no compression occurs in this stage, the authors investigate the effect of various mother wavelets, and decomposition structures on the overall system performance. Specifically, they compare the ubiquitous Daubechies 9/7, the Villasenor 10/18 and a 28/28 biorthogonal filter. In addition they compare the standard dyadic decomposition with the packet decomposition.

	Lena – PSNR (dB)			Goldhill – PSNR (dB)		
Filter	7/9	10/18	28/28	7/9	10/18	28/28
0.25bpp	34.66	34.77	34.76	30.90	30.96	30.98
0.5bpp	37.79	34.87	37.87	33.68	33.73	33.75
1bpp	41.10	41.12	41.21	37.17	37.29	37.34

Table 3.12: Effect of Wavelet Filter on TCSFQ [Xiong99,2]

Results show, as in Section 3.2.1.1, that performance increases with filter length, which is at the cost of increased computational burden. Again this improvement is at best marginal, being on average 0.1dB PSNR.

Also investigated is the effect of the packet decomposition on the notorious Barbara image, which is heavily textured. Technically the decomposition is the 3+3 decomposition that uses a full packet decomposition for the first 3 levels, and is dyadic for the remaining 3.

	Barbara – PSNR (dB)					
Transform	Dyadic			Packet		
Filter	7/9	10/18	28/28	7/9	10/18	28/28
0.25bpp	28.55	28.81	29.17	29.61	29.87	30.60
0.5bpp	32.57	33.00	33.35	33.36	33.83	34.48
1bpp	37.71	38.25	38.46	38.15	38.48	39.12

Table 3.13: Effect of Packet Decomposition on TCSFQ [Xiong99,2]

The result agrees with those discussed in 3.2.1.3. In that section, optimal packet pruning returns 3-4dB gain (on 'Barbara'), while using this unpruned packet decomposition yields 0.5-1.5dB.

3.3.10.3 Quantisation

Xiong's own previous SFQ algorithm (Section 3.3.5) showed that optimal zerotree pruning based on RD criteria yielded the best results for zerotree quantization. Furthermore, the Markovian nature of wavelet coefficients is not exploited in this method. This motivates the use of trellis coding (3.3.7) as a complementary quantization mode. Xiong thus replaces the original uniform threshold scalar quantiser (UTQ) of SFQ with a trellis coded quantizer (TCQ), for coding those coefficients not pruned by the zerotree coding stage.

The results of this substitution are shown in Table 3.14.

	PSNR (dB)	
Rate (bpp)	SFQ+UTQ	SFQ+TCQ
0.25	34.12	34.76
0.5	37.18	37.87
1.00	40.74	41.21

Table 3.14: Effect of Trellis Coding in SFQ [Xiong99,2]

It should be noted that these are the results for the SFQ coder, without any context modelling in the entropy coding stage.

The improvement in results is due to the enlarged codebook the TCQ allows, as a result of exploiting the Markovian characteristics of the wavelet domain.

3.3.10.4 Entropy Coding

The final stage is the entropy coding stage, here recent advances such as in ECECOW, motivate the use of higher order context formation. By inserting the ECECOW entropy coder as the final stage into the original SFQ algorithm (SFQ+ECECOW), and the trellis coded amendment to SFQ (SFQ+TCQ+ECECOW) the following performance is exhibited:

	PSNR(dB)		
Rate (bpp)	SFQ	SFQ +ECECOW	SFQ+TCQ+ECEOW
0.25	34.22	34.77	34.76
0.5	37.36	37.85	37.87

1.00	40.52	41.13	41.21

Table 3.15: Effect of Context Formation on SFQ

Interestingly, the performance gain of ECECOW in addition to TCQ is minimal. The reason is that both attempt to exploit the same redundancy; the Markovian dependency between spatially neighbouring coefficients. As ECECOW uses a higher order model than TCQ, and is more adaptive to the underlying data; it produces better results than TCQ.

3.3.10.5 Discussion

[Xiong99,2] presents several important results. The first being, that the zerotree quantization method is independent of the Markovian behaviour of images. Thus, exploiting this statistical property presents an opportunity to increase performance. As ECECOW. utilises the most sophisticated Markov modelling, it exhibits the greatest benefit.

An interesting result is that ECECOW[Xiong97] outperforms TCSFQ, even without the advanced context quantisation options added in 2001 [Xiong01]. This is not explored in the paper, which fails to compare TCSFQ with ECECOW. However, I believe that this is as a result of improper application of ECECOW in this work. It appears from the paper that the authors applied ECECOW directly to the already zerotree processed coefficient set. As the zerotree coding significantly alters the image statistics, such as by flattening the distribution around zero, the original ECECOW context model may not hold. In order to fully exploit the capabilities of ECECOW, the exercise of reforming the context modeller would have to be undertaken on a suitable sample of zerotree processed coefficients. This must be done to reveal the statistical properties of this coefficient set, which will be different to the original wavelet coefficient set.

The computational requirements of this method are unpublished, however given the nature of the algorithm, will be considerable.

3.3.11 JPEG 2000 and EBCOT

The previous decade of activity in image compression using wavelets and the resultant performance gains motivated the standards bodies, the ITU and ISO, to update the DCT based JPEG algorithm to incorporate wavelet techniques. The resulting scheme is known as JPEG2000 [Boli00], [Chri00]. The system is complex, with many usability and error resiliance features. However, as these are ancillary to our discussion, the material here shall be limited to the core coding scheme, Embedded Block Coding with Optimized Truncation, EBCOT, [Taub00].

3.3.11.1 Premise

EBCOT presents the joint application of many of the techniques presented in isolation in previous sections. The central coding features are an adaptive binary arithmetic coding scheme driven by a bit-plane context formation scheme, coupled with an RD optimisation scheme based on bit rate allocation within image tiles. In addition, the output stream is carefully constructed to be SNR and resolution scalable.

3.3.11.2 Algorithm

After the image has been wavelet transformed, EBCOT employs a two stage algorithm to code the coefficients.

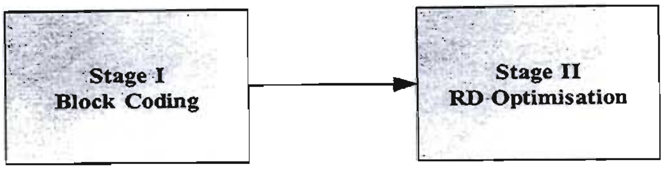


Figure 3-12: EBCOT Coding Stages[Taub00]

3.3.11.2.1 Stage I: Block Coding

The first stage tiles each subband into blocks, usually dimension 16x16. Each of these blocks is then coded separately to produce an embedded stream. The coding follows a very similar process to ECECOW (3.3.8); an adaptive binary arithmetic coder, driven by bit-plane context formation.

Each tile is traversed in a raster scan, bitplane order.

Context Formation

Each of the three coefficient properties, significance, sign and magnitude, has an independent context. This is done to reveal the statistical properties unique to each property.

The significance-, termed Zero-, coding refers to the probability of a previously insignificant coefficient becoming significant at the current bitplane. This property was found to be Markovian, as in (ECECOW, 3.3.8); dependant only on the significance of its eight spatial neighbours. Interestingly Taubman only includes spatial neighbours, and not parent or child neighbours as in ECECOW.

As there are 256 possible conditioning states, the context dilution problem is encountered. Thus, the number of states is reduced by considering the following states, based on the number of neighbours in a given horizontal (H), vertical (V), or diagonal (D) orientation:

# coeff			
H	V	D	State
0	0	0	0
0	0	1	1
0	0	>1	2
0	1	X	3
0	2	X	4
1	0	0	5
1	0	>0	6
1	>0	X	7
2	X	x	8

Table 3.16: Significance Context Formation in EBCOT[Taub00]

This table only indicates the contexts for the LL, HL and LH subbands; a separate table is used for the HH subband, as the subband correlation structures differ. As the table indicates, additional states are expended on encoding the horizontal dimension, as the correlation structures are strongest in this direction in the LH band. In EBCOT the HL band is transposed before this context formation to allow its vertical structures to be captured with the same contexts. This band dependant context formation is a similar approach to that taken by ECECOW.

If a run of greater than 4 insignificant coefficients is found, it is encoded using run-length coding rather than the coding described above.

The sign information is encoded in a similar manner. Taubman argues that in LH (and transposed HL bands) horizontal neighbours tend to share signs, while vertical neighbours have opposite signs. This is owing to the wavelet filter operating in the horizontal direction being of type low-pass thus outputting a continuous sign due to the input of typically low pass image data, and the vertical filter tending to do the opposite. This argument is supported by empirical evidence presented in [Wu01]. A context formation table is constructed to take advantage of this correlation.

Finally the magnitude context is formed. The correlation structures for this property have yet to be identified. The conditioning states for this variable are only whether the coefficient has previously become significant, and whether any horizontal or vertical neighbours are significant.

Encoding

Having formed the context for each coefficient, a specific scanning order is followed to input the coefficient and its context into the arithmetic coder. This is done to encode the most significant coefficients first.

In summary, first the coefficients with significant horizontal neighbours are coded (owing to the horizontal correlation mention previously), then the coefficients with at least one significant neighbour, then the coefficients found significant at a previous bit plane, and finally the remaining coefficients.

The arithmetic coder entropy codes this data, producing an embedded output stream.

3.3.11.2.2 Stage II: RD Optimisation

As each tile has been coded in an embedded fashion, it is possible to assign the bitrate optimally between the tiles by truncating the output stream of each tile, hence the nomenclature, optimal truncation.

During the block coding stage described above, the set of possible truncation points for the scheme is generated. At each point the rate, and slope of the R(D) curve is internally stored as auxiliary data. This stage produces the final output stream by using this information and a Lagrange multiplier method to allocate the rate between the tiles in an RD optimal manner.

The Lagrange multiplier method is unremarkable except for the distortion metric used. Rather than perform RD estimation in the original spatial domain, the following distortion metric is used in the wavelet domain:

$$\hat{D}_i^n = w_{bi}^2 \sum_{\underline{k}\in B_i} \left(\hat{s}_i^n\left[\underline{k}\right] - s_i\left[\underline{k}\right]\right)^2 \quad , \tag{3.21}$$

where $\hat{s}_i\left[\underline{k}\right]$ represents the 2D set of wavelet coefficients forming block B_i. Similarly, $\hat{s}_i^k\left[\underline{k}\right]$ represents the same coefficient set, truncated to point n. Finally, w_{bi} is the L2 norm of the basis function used in the wavelet decomposition. This metric approaches the Square Error distortion metric in the spatial domain, if the basis functions used are orthogonal. However, they are only quasi-orthogonal, thus this metric is by definition inaccurate. Taubman motivates its use for simplicity; not having to transform the coefficients back to the spatial domain in order to gain a distortion metric.

Thus, for each tile a set of points is defined on the R(D) curve. An operational RD optimisation approach is then taken, as explained in Chapter 2, to allocate the rate optimally between the tiles.

3.3.11.3 Results

The following results are reported for the algorithm:

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	34.29
Lena	0.5	37.41
Lena	1.0	40.57
Barbara	0.25	28.51
Barbara	0.5	32.43
Barbara	1.0	37.37

Table 3.17: EBCOT Performance Results[Taub00]

Unfortunately JPEG2000 literature uses a different test image set to that used in prior literature, making a direct comparison between JPEG2000 and the other algorithms presented impossible. However, by comparing EBCOT and JPEG2000, and then using the above results, insight may be gleaned.

Image	Bit Depth (bpp)	JPEG2000 PSNR (dB)	EBCOT PSNR (dB)
Café	0.25	23.06	23.26
Café	0.5	26.42	26.97
Café	1.0	31.9	32.24

Table 3.18: Comparison between JPEG2000 and EBCOT [Taub00],[Chris00]

The above results show that EBCOT represents the limiting performance of the JPEG2000 algorithm. This is because JPEG2000 implement EBCOT at its core, but adds many features such as ROI coding and an error resilient output stream among others. This adds rate overhead, reducing performance relative to EBCOT.

3.3.11.4 Discussion

EBCOT is the most modern method presented in this chapter. It combines many of the ideas of previous chapter to produce a complex scheme. Although the results presented are inferior to many of the simpler algorithms, this is owing to the framework within which EBCOT has been produced.

Not mentioned in the above discussion is an output stream layering method, which combines the output of each block into a SNR and resolution scalable stream. Furthermore, many configuration options, such as block-size, number of output stream layers, quality metrics and others, must be indicated in the output stream. These features are necessary as part of the JPEG2000 framework, but are RD expensive. The inferior performance, relative to algorithms that do not carry this overhead, is thus expected.

JPEG2000 is designed to be a generic image compression algorithm, supporting many features (in addition to those of EBCOT) and possessing error resilient properties. These are necessary as the scheme is intended for general and widespread use. These extra properties however, come at the cost of compression performance.

3.4 Summary of Results and Discussion

The results found in the preceding chapter will be collated and discussed, according to the stage of the compression system.

3.4.1 Wavelet Transform Stage

It was found that the degrees of freedom included the mother wavelet, the number of mother wavelets, the dimensionality of the mother wavelet, and the decomposition structure.

Altering of the mother wavelet was found to yield at most 1dB of PSNR improvement. However, gains beyond the performance of the standard Daubechies 9/7 were neglible. Projection onto a system of multiwavelets was usually found to produce inferior, or at best comparable, performance to the standard scalar case. Truly two dimensional basis functions are in the process of development, but are still too new an avenue for good quantisers to exist to code their output, thus no real results are available.

Altering the decomposition structure using a packet decomposition, followed by an optimal tree pruning algorithm yields substantially enhanced performance, of up to 5dB PSNR. This performance is only demonstrated in the case of a complex, highly textured image, 'Barbara' and it is probable that similar gains will not be achieved for more usual images.

Thus for the purposes of this project, there are no gains to be made from departing from the standard dyadic decomposition structure using the Daubechies 9/7 wavelet.

3.4.2 Quantisation Stage

The first group of wavelet image coders considered are those that perform the statistical modelling implicitly:

		PSNR (dB)				
Image	Rate (bpp)	JPEG	EZW	SPIHT	SFQ	SR
Lena	0.25	31.6	33.17	34.11	34.33	33.87
Lena	0.5	34.9	36.28	37.21	37.36	36.93
Lena	1	37.9	39.55	40.46	40.52	40.23
Barbara	0.25	25.2	26.77	27.79	28.29	28.90
Barbara	0.5	28,3	30.53	31.72	32.15	32.66
Barbara	1	31	35.14	36.41	36.86	X

Table 3.19: Comparison of Implicit Model Coders

Except for SR they all employ the zerotree quantisation mode. The implicit assumption is that energy is localised in the low frequencies in the wavelet domain, and that high frequency energy is caused by edges and thus is spatially localised. The zerotree structure takes advantage of both of these assumptions, as explained in the text.

EZW is the original work in the field, the performance of SPIHT improves on this generating the output stream more efficiently. Specifically by descending a coefficient tree to output high value child coefficients, before moving to the next coefficient, SPIHT approaches the R(D) convex hull more closely than EZW. SFQ argues that the zerotree quantisation utilised in EZW and SPIHT is suboptimal in an RD sense. This is because the rate cost of not zerotree encoding a low energy tree is not considered, only the distortion cost. Argued differently SPIHT always behaves so as to minimise the distortion, without proper regard to rate implications. SFQ thus considers the RD contribution of each tree and zerotree quantises them in an optimal fashion.

SR coding is designed for low bitrate transmission where the number of significant coefficients will be small. Thus, rather than employ zerotree coding, this method uses runlength coding to capture large strings of insignificant coefficients within the subbands.

The second set of methods relies on explicit statistical modelling in the wavelet domain.

		PSNR (dB)			
Image	Rate (bpp)	TCQ	EQ	ECECOW97	ECECOW01
Lena	0.25	34.43	34.57	34.81	34.89
Lena	0.5	37.69	37.38	37.92	38.02
Lena	1	41.47	40.88	40.85	41.01
Barbara	0.25			28.85	29.21
Barbara	0.5			32.69	33.06
Barbara	1			37.65	38.05

Table 3.20: Comparison of Explicit Model Performance

The performance gain of these methods relative to the implicit model codecs is evident. TCQ and ECECOW are based on modelling the wavelet coefficient set as Markovian. TCQ models each coefficient as dependant on only its raster scan predecessor. By using the trellis structure TCQ expands the number of quantisation levels available. This is achieved by making only a subset of the quantisation levels available to a given coefficient, based on its context (value of previous coefficient), and coding these levels with symbols, whose meaning varies according to the context.

ECECOW presents a more sophisticated Markovian conditioning. Strictly speaking it is not actually a Markov model, but as the conditioning states consist of coefficients in very close proximity to the one of interest, it has become practice to refer to the method as Markovian. A linear best estimate for the value of a coefficient is produced, based on the surrounding pixels. This neighbourhood is selected to reflect subband specific properties that are known to exist. This expected value is quantised, and combined with the significance of the coefficient's neighbours to form a probability estimate of the pixel. This estimate is input to an arithmetic coder. The later version of ECECOW includes a sophisticated context quantisation methodology.

EQ uses "mixture modelling." Each wavelet subband is modelled as the superimposition of Generalised Gaussian Distributions centred around each coefficient. The variance, and shape parameter of these distributions are estimated adaptively online. Based on these estimated distributions, each coefficient is quantized according to optimally matched quantization tables, which have been designed offline.

It can be seen that the application of explicit statistical modelling yields returns in compression performance. This is because the assumptions of zerotree quantisation do not always hold, and the greater flexibility of these later schemes allows them to adapt to the image properties.

Noting that several of the methods presented above operate by exploiting different properties of the wavelet domain, some new techniques are presented which combine various methods into one algorithm.

The most successful of these methods is the TCSFQ algorithm. This combines the RD optimal zerotree quantization method, with trellis coding of the significant coefficients, followed by ECECOW based context based arithmetic coding. This algorithm presents good results:

	PSNR (dB)		
Rate (bpp)	SFQ	SFQ +ECECOW	SFQ+TCQ+ECEOW
0.25	34.22	34.77	34.76
0.5	37.36	37.85	37.87
1.00	40.52	41.13	41.21

Table 3.21: TCSFQ Results

The contribution of the trellis coding (TCQ) is seen to be minimal, when considered jointly with ECECOW. This is because the Markovian redundancy exploited by TCQ is already accounted for in ECECOW. Furthermore the TCSFQ algorithm is outperformed by ECECOW; one of its constituents. This may be because the ECECOW contexts are designed based on natural image data, whereas ECECOW is being applied to already zerotree quantised data in this case. Thus the ECECOW modelling will be suboptimal in this case. Redesigning the ECECOW context formation for the zerotree quantised source should yield performance returns.

The final algorithm considered is the JPEG2000 coding engine, EBCOT.. The core of this algorithm is a simple context driven arithmetic coder. The context is formed from each coefficient's immediate spatial neighbours (unlike ECECOW that also considers parent-child relationships). Each wavelet subband is segmented into tiles, and each one independently coded with this arithmetic coder, which produces an embedded output stream for each tile. A Lagrange based RD optimisation is undertaken by truncating each tile output stream to minimise the overall distortion.

The performance results of this algorithm are presented below:

Image	Bit Depth (bpp)	PSNR (dB)
Lena	0.25	34.29
Lena	0.5	37.41
Lena	1.0	40.57
Barbara	0.25	28.51
Barbara	0.5	32.43
Barbara	1.0	37.37

Table 3.22: EBCOT Performance Results

Despite the sophisticated algorithm, these results are not the best. This is possibly owing to the simple context model used, as well as the sundry features of the output stream. EBCOT is used as the core of the JPEG algorithm, and this supports many configuration options and output stream variations. This configurability imposes a rate overhead that may explain the results.

JPEG2000 is based on the EBCOT algorithm, and adds many more features such as error resilience and region-of-interest coding. These add further rate overhead, reducing the RD performance from EBCOT. However, given the widespread commercial application of JPEG2000 this cost is acceptable.

3.5 Conclusion

This chapter has presented the leading ideas for still image compression using wavelet techniques.

In the wavelet transform stage it has been found that modifying the basis function itself is of very little utility. The only reported gains are for coding complex images, and using adaptive packet transforms.

In the quantization and entropy coding stages, ongoing gains are being made. It was found that the underlying model used is the determining factor in performance. The original subband model was improved to that used by zerotree coders. Recently that model has been found restrictive, and newer, flexible models have been proposed. The performance gains of these carefully designed, adaptive models are readily apparent.

Recognition of the importance of source modelling motivated the work of Chapter 5, in which a video source modelling exercise is undertaken.

Chapter 4 - Wavelet Video Compression

Having examined the compression of still images in Chapter 3, this chapter proceeds to the next level of complexity; video coding. Video may be considered as a source consisting of a sequence of still images, or frames. Each of these frames has spatial correlation structures similar to normal images. Consequently, many image coding ideas are borrowed and extended for coding these structures.

Video sequences commonly run at between 10 and 30 frames per second (fps). As a result, not much usually changes between one frame and the next. Hence, spatial regions in one frame are highly correlated with spatial regions in other frames that occur slightly before or after the frame in question. This property is referred to as temporal correlation, and a major effort of video coding is to discover and describe the correlation structures.

The process of video compression consists of the joint exploitation of the spatial and temporal correlation structures.

Changes in the frame may be a result of camera motion, such as zooming, panning, rotating or tilting. This mode of motion affects every pixel in the frame, hence is referred to as global motion. Several specific algorithms attempt to detect and compensate for these effects. Other changes may be a result of object motion, such as translation, rotation, occlusion, uncovering or morphing. These motions are restricted to the spatial region of the object undergoing motion, and are hence referred to as local motions. The underlying strategy for dealing with both local and global motion is motion estimation and compensation (ME/MC). For instance a general process would be to compare two frames and estimate the motion from the first frame to the second. This motion is then artificially introduced to the first frame by the algorithm; this is called motion compensation. The difference between this motion compensated frame and the second frame is called the error, or residual frame. The video encoder then only has to communicate the motion estimate, and residual frame, which has proven highly effective, as will be shown in coming section.

Other ideas have been presented in competition to the ME/MC paradigm. Recently systems that exploit the temporal redundancy directly with signal processing devices similar to those employed in still image coding have been proposed. Most of these schemes are referred to as 3D systems as they employ the same decorrelation method in both spatial directions, and the temporal direction. These systems will also be explored below.

Finally, any other changes are a result of lighting changes or noise, these are generally unpredictable effects.

This chapter will open with Section 4.1 that presents the existing video coding standards of ISO and the ITU. These are all based on ME/MC and the discrete cosine transform. This discussion is presented to illustrate the state of the art, as a background to the later discussion of wavelet video compression.

The new wavelet video coding methods are presented in Section 4.2. Many of these methods borrow concepts and techniques from the existing standards, another reason for the inclusion of Section 4.1. Other methods are based on entirely new concepts made possible by wavelets. The discussion is structured according to the method of temporal decorrelation, and in each case one or two algorithms are presented, with comparative performance results.

Finally a discussion and conclusion are presented in Section 4.3.

4.1 Video Compression Standards

In order to provide context to the new wavelet techniques for video compression, it is useful to briefly analyse the existing standards. There are two main standardising efforts; the Moving Pictures Experts Group (MPEG) which exists under the broader ISO committee, and the H.263 effort that is run by the ITU. This section will briefly discuss the enabling concepts from these standards. Much of this discussion is drawn from [Effe98], [Bhas97] and [Sola97].

The following discussion is intended only to introduce the concepts used in the existing standards, and does not present a review of either algorithms or results. Results are presented in the wavelet video coding section by means of comparison.

4.1.1 MPEG 1

MPEG-1 [MPEG91] is the original MPEG standard. It was developed primarily for the storage of video onto media such as DAT, and as such is optimised for a bitrate of approximately 1.5Mbit/s.

Interestingly, the MPEG standards (1, 2 and 4) only specify the syntax of the coded stream and not the implementation details. This is to foster competition and allow developers to find and adopt new techniques, thus maintaining the relevance of the standard. However, the syntax does imply the underlying strategy.

MPEG-1 is based on block motion estimation and motion compensation (ME/MC), followed by discrete cosine transformation (DCT) of each block, scalar quantisation, and run length entropy coding.

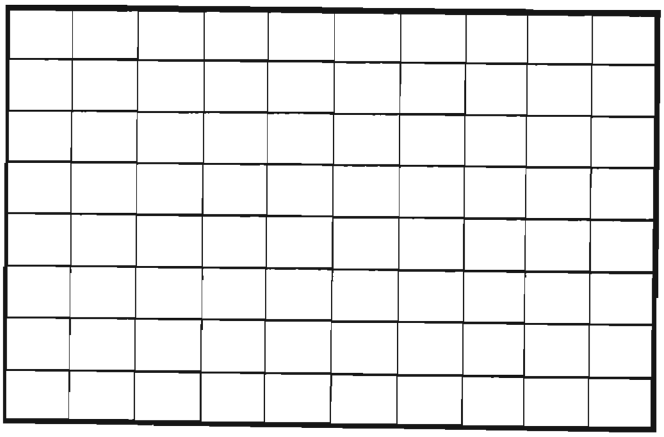


Figure 4-1: MPEG-1 Frame Blocking

Figure 4-1 above demonstrates the blocking of a frame. Each luminance frame is broken into 16x16 pixel blocks, which are referred to as macroblocks.

The exploitation of temporal correlation is achieved through ME/MC. The assumption is that most of the changes between frames are due to motion of the bodies in the scene. The texture of these bodies does not change significantly between frames, only the spatial location. Thus, in forward predictive coding, blocks from the previous frame are matched with a block in the current frame. The displacement of a block is referred to as a motion vector (MV). More advanced block matching considers blocks in both previous(forward estimation) and subsequent(backwards estimation) frames, in a group of frames (GOF) of several frames, to form an interpolated vector. Thus the MV for each block (should it exist) is output to the stream, as well as the error between the predicted block and the actual block. This dramatically reduces the information content of the sequence, and is the main element of temporal compression in MPEG.

The spatial decorrelation is performed next. The error between the predicted block and actual block is coded in 8x8 blocks using the DCT, as in JPEG [JPEG], scalar quantized and then entropy coded using run-length coding. As these are not the subject of current enquiry they will not be explored, however Chapter 3 on still image coding explores some of these ideas.

A block diagram of the MPEG-1 encoder is shown in Figure 4-2.

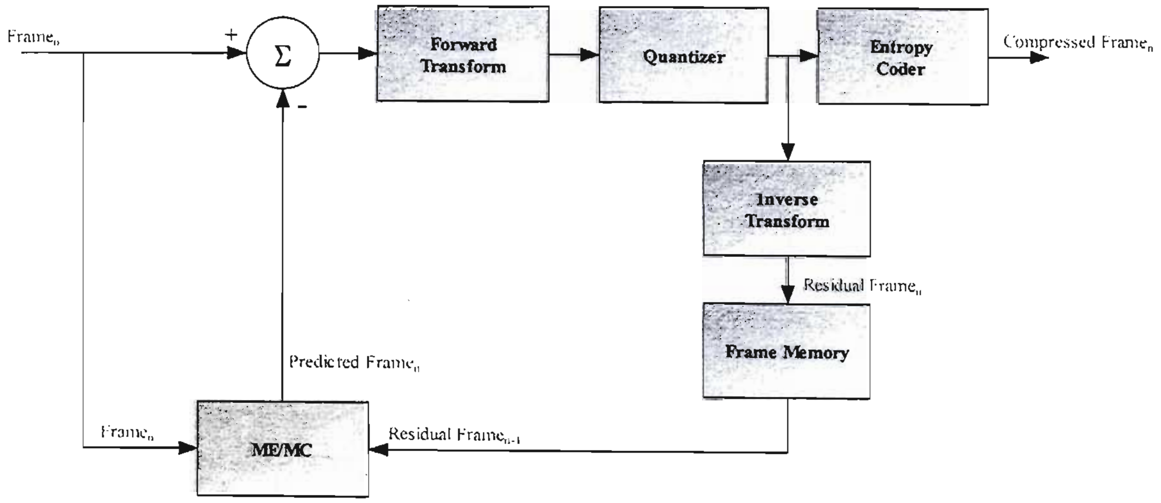


Figure 4-2: MPEG-1 Block Diagram, based on [Effe98]

In the block diagram, motion estimation refers to the process of generating the motion vectors, and motion compensation, the construction of a predicted frame based on these motion vectors. The output of the summing block is the difference between the actual frame, and that produced by the motion estimation scheme, and is referred to variously as the error- or residual-frame.

It is noted that the major contribution of the MPEG-1 algorithm is block ME/MC, as it is really this process that defines the MPEG algorithm differently from JPEG.

4.1.2 MPEG 2

MPEG-1 was designed for the storage of video. MPEG-2 [MPEG93,2] is designed for the broadcast of video over terrestrial and satellite TV, and storage onto devices such as DVDs. For this reason MPEG-2 includes many stream management features to increase error resilience and delivery. While MPEG-1 is optimised for a bandwidth of 1.5Mbit/s, MPEG-2 is designed to handle video up to about 100Mbit/s for applications such as high definition television (HDTV). In addition various features, such a variable resolution, SNR and frame rate are incorporated into MPEG-2. As the application of MPEG2 is towards television, various additions are made to handle interlacing.

The actual video coding is very similar to MPEG-1 (MPEG-2 is backwards compatible with MPEG-1). The coding is based on block ME/MC followed by scalar quantisation and Huffman entropy coding. Additions include:

- ME/MC modes for interlaced video for TV applications
- Partitioning of 16x16 macroblock sizes into two 16x8 blocks during ME/MC
- A new motion compensation algorithm

4.1.3 MPEG 4

MPEG-4 [MPEG-4,2], [MPEG-4,3] is the current incarnation of the MPEG standard. It is intended for the compression of multimedia scenes. Such scenes include computer generated scenes, such as virtual reality, animations and synthetic video. It is also able to handle multiple camera scenes. Unlike previous coding standards it is an object rather than a signal based algorithm. The input to an MPEG-4 codec is a scene, a background with various foreground objects such as sprites, avatars and text. In addition, music, speech and other components of a multimedia presentation are considered objects. The motivation for this paradigm shift is to encourage interactive applications, noting that humans interact with objects and not signals.

Restricting the discussion to the video aspect of the codec, there are two main aspects of coding; that of objects, and that of raw video. There is much research activity in the field of object segmentation; the extraction of a meaningful object, such as a person, from a raw video sequence. An interesting note is that part of the standard defines the coding of object textures using a wavelet transform, this is the first use of wavelets in MPEG. This aspect of object coding is beyond the scope of this thesis.

The coding of unsegmented video is also considered in the standard, this is included to allow advantage to be taken of advances in video coding since MPEG-2. Although the video may be coded according the block ME/MC and DCT much like MPEG-2, there are several new options. These new features are mainly with regard to ME/MC. New global motion estimation schemes are presented, based on either affine transformation of the entire frame or just the background.

4.1.4 H.261 and H.263

Operating almost in parallel with the MPEG effort from ISO, is H.261 and H.263 standardisation process from the ITU [H263,2]. These standards focus on low-bitrate video coding for transport over integrated services digital network (ISDN) and dialup internet links respectively, for video telephony.

H.261 codes data with an ouput rate of n*64kbits/s, this is because the ISDN channel consists of multiple 64 kbps lines. H.261 is extremely similar to MPEG-1, and operates on the same principles, thus, here, only the significant differences will be given.

The most significant difference is the handling of ME/MC in H.261. As with MPEG-1, macroblocks of 16x16 are considered for ME. However, only the previous frame is used to generate motion vectors, rather than a group of frames as in MPEG-1. In addition, the search area used to find a matching macroblock is limited to ± 15 pixels, while in MPEG-1 the entire frame may be used. These limitations may be seen as disadvantageous; however they are required given the application. In MPEG-1 the intended use is offline recording of video, thus computational delay is not critical. However, H.263 targets real time video conferencing and as such large computational delays are unacceptable, this motivated for these limitations.

Other less important differences are that H.261 uses Huffman entropy coding, and uniform scalar quantization, while MPEG-1 uses run length coding, and different scalar quantisation stepsizes for each frequency.

H.263 is an extension of H.261, intended for very low bandwidth video, such as 20kbits/sec. The major changes from H.261 relate to ME/MC. Half pixel accurate motion vectors are a new feature, as is the removal of the range limitation on the motion vector search, and the reduction of the macroblock size to 8x8. Also included is backward motion estimation, as in MPEG. All these features are to be found in the MPEG 1 and 2 standards. Original features, later incorporated into MPEG-4, include overlapping block matching to handle occlusion and deformation of objects, motion vectors that may extend out of the picture to efficiently account for objects that have moved out of the frame, and arithmetic coding.

4.1.5 Summary of Video Coding Standards

This section has presented the major standards that currently exist for the coding of video.

It is quite apparent that in every case the temporal decorrelation is performed by motion estimation and compensation. Following this process spatial coding is achieved using the DCT and quantization. Subsequent versions of the ISO and ITU standards really represent refinements of the process for different usages. MPEG-4, however, does represent a new departure with the focus on objects rather than signals. These standards provide a useful comparison for the wavelet techniques that follow, both in terms of their concepts and performance.

4.2 Proposed Wavelet Methods

Wavelet based schemes have long outperformed the classical DCT based schemes in still image compression. This is well evidenced by the latest JPEG standard, JPEG2000 [JPEG200,1], which is based on the wavelet transform. Current research is aimed at repeating this success in the coding of video. This field of endeavour is still young and limited success has been achieved.

It is not surprising that existing video compression schemes rely on the DCT, as existing image compression routines do as well. It is well known that the DCT requires the image to be segmented into blocks. These blocks provide a natural means of temporal decorrelation, through block matching motion estimation and compensation. This technique has proven highly effective, as indicated by its usage in the standards presented previously. These DCT based codecs have for a long time outperformed the competing wavelet video codec offerings.

However the latest research has produced wavelet schemes based on several different techniques that compete and surpass MPEG and H.263. Significantly, many of them offer the fine rate and frame control and other 'value-added' properties that are important in applications. Of particular interest is the rate control, which allows a realtime video coder (such as that used for video teleconferencing over the internet) to react to changing network conditions. This issue is becoming increasingly important as the uptake of this technology continues.

There is no wavelet video compression standard yet. The sections that follow will present an overview of the research in the field of wavelet video compression. The discussion will draw from Chapters 1, 2 and 3, as many of the ideas are shared.

As this field is still rather young there are many different schools of thought. Section 4.2.1 will attempt categorize the methods according to their operating principle. This is done to provide some structure to the review of methods that follows.

Sections 4.2.2 through 4.2.5 will review many important methods from the literature. It will proceed according to the structure laid out in Section 4.2.1. This section will not only present the most competitive methods, but will also include methods that represent different aspects of wavelet video coding.

Finally Section 4.3 will compare and discuss the methods on their merits.

4.2.1 Classification of Methods

The current wavelet video codec offerings may be classified into four broad categories:

- Intraframe Coding
- 3D Wavelet Coding
- Spatial Domain ME/MC
- Wavelet Domain ME/MC

The following discussion will first define each of these categories briefly. Thereafter, leading work in each category will be more thoroughly explored. Each method will be discussed with reference to the following diagram, which shows a generalised video coder block diagram.

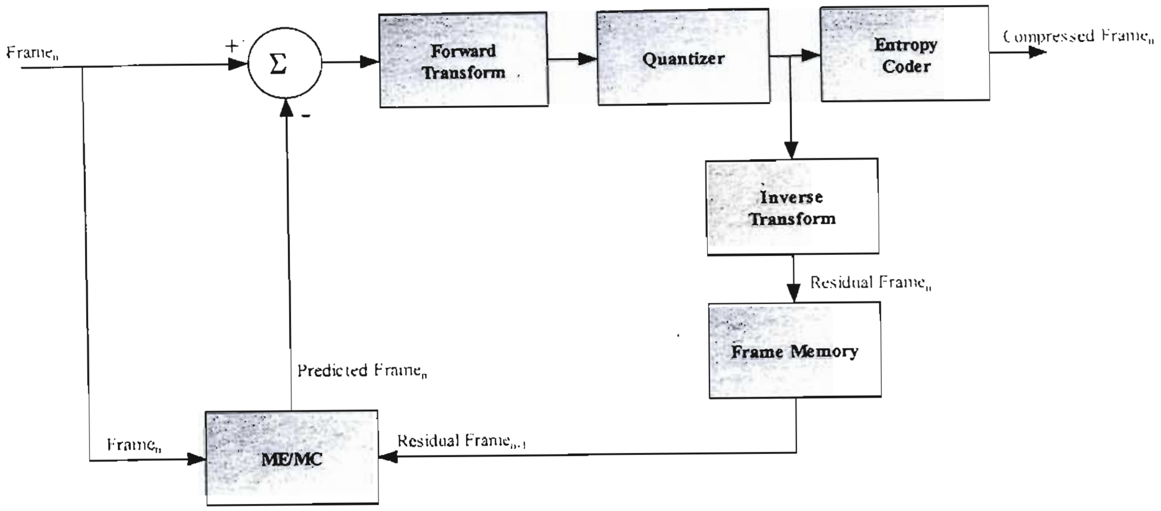


Figure 4-3: Generalised Video Coder Block Diagram

4.2.1.1 Intra-frame Coders

This family of techniques applies known still image compression techniques independently on a frame-by-frame basis to a video sequence. These are not true video codecs as no consideration is taken of the temporal correlation between frames. These algorithms only make use of the forward path of the video coding diagram:

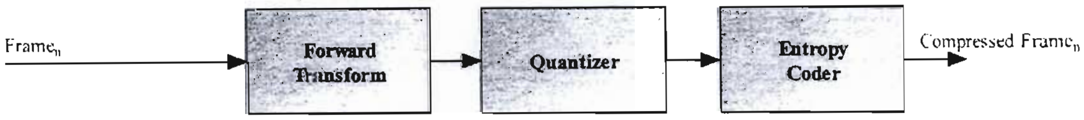


Figure 4-4: Difference Frame Block Diagram

4.2.1.2 Difference Frame Coders

The simplest form of temporal decorrelation is difference coding. Under this scheme, the difference between the current and previous frame is encoded. It may be thought of as a ME/MC scheme where all the motion vectors are assumed to be zero, thus only the error information is encoded.

In terms of the block diagram, these algorithms make use of the feedback path, but without ME/MC.

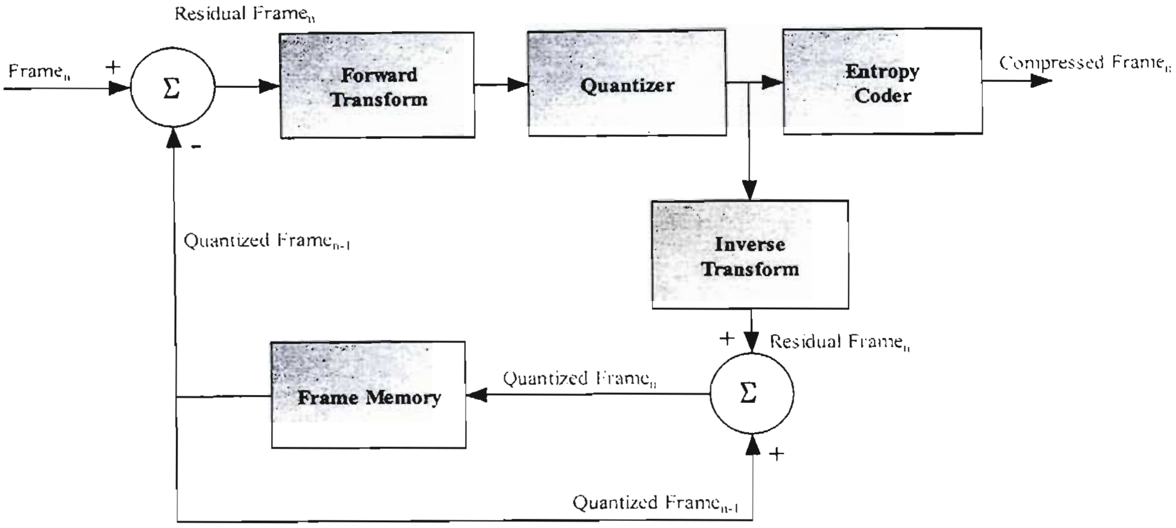


Figure 4-5: Difference Coder Block Diagram

4.2.1.3 Spatial Domain ME/MC

It has been mentioned that the currently employed schemes of MPEG and H.263 draw their performance from their efficient motion modelling. Briefly, this is the formulation of a video frame as a quasi-static background with multiple moving rigid bodies forming the foreground. These schemes aim to encode the background and object separately, and transmit only the motion of the objects. It is evident that significant compression will result, should this be achieved. These coders utilise the generalised block diagram of Figure 4-3, as shown.

There are two current philosophies to attain this end; block- and object-based ME/MC.

4.2.1.3.1 Block Based ME/MC

Several wavelet video coding techniques employ nearly identical ME/MC algorithms as discussed in Section 4.1. However, as block ME/MC requires the image to be fragmented into blocks, it naturally suffers from the infamous 'blocking-effect' in a low bandwidth scenario. This visual defect was a major motivation for the still-image compression community moving towards the wavelet transform, and there is thus reticence to employ block based methods in future wavelet video standards. Furthermore, block ME/MC is a natural method given the blocking requirements of the DCT, however for wavelet techniques this blocking is an artificial imposition.

4.2.1.3.2 Object Based ME/MC

An extremely active field of research attempts to determine which pixels in a video frame correspond to physical objects, in order to encode them separately to the background. This idea was explored in Section 4.1.3. Indeed as mentioned there, the wavelet transform is proposed for the coding of object textures. If and when the object extraction problem has been satisfactorily solved this may well be the form that wavelet video compression will take. At this time however the problem is not solved, thus other methods are being explored.

4.2.1.4 Wavelet Domain ME/MC Methods

All the above ME/MC techniques operate in the spatial domain. Each frame is encoded, and then inverse coded before ME/MC occurs (Figure 4-2). New techniques are being developed that operate within the wavelet domain and aim to encode motion efficiently.

Pure wavelet domain ME/MC is complicated by the extreme aliasing found within the wavelet domain. In general the extent of signal processing in the wavelet domain has been the removal of low magnitude coefficients, either for the purpose of image de-noising or compression, due to this problem. Amelioration techniques have been found to limit the effect, and a group of techniques known as hierarchical motion estimation has been developed. Hierarchical motion estimation uses the coarse resolution wavelet subband to perform initial motion estimation. As this subband contains the least number of coefficients, this is efficient. Thereafter, this motion estimation is used to predict motion in each subsequent higher frequency band.

The block diagram for these methods takes the form shown in Figure 4-6.

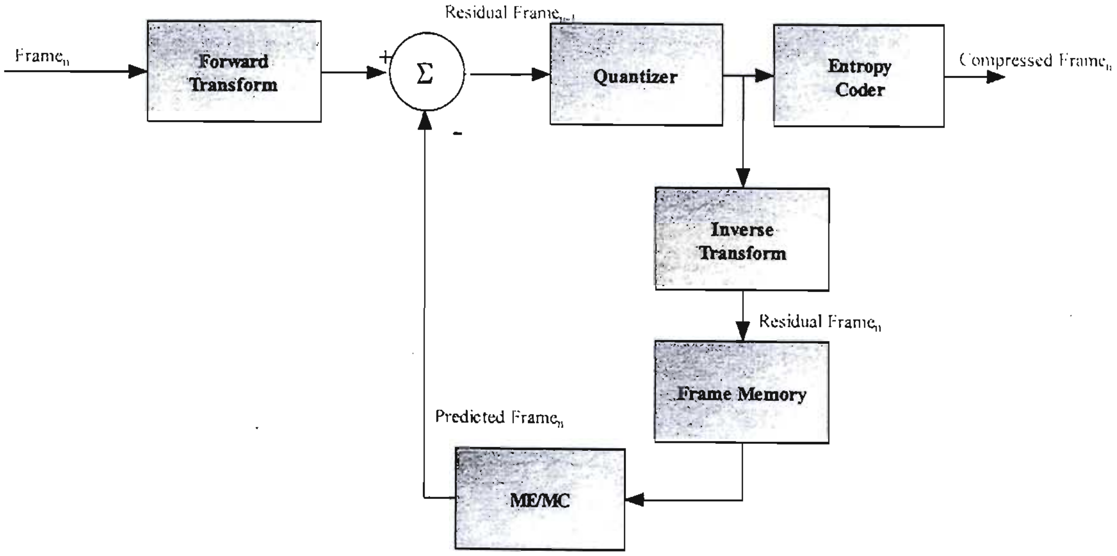


Figure 4-6: Wavelet Domain ME/MC Block Diagram

4.2.1.5 3D Wavelet Coding

There are two classes of techniques within the broad category of schemes that are aware of the temporal dimension; those based on motion estimation, and those that are not. 3D Wavelet coding is of the latter grouping, and will be defined below.

These schemes extend the 2D algorithms such as SPIHT and SFQ to include the temporal dimension. In general, a group of frames, GOF, is buffered in memory and the wavelet transform is performed independently in each of the three directions, x, y and t. Then these coefficients are compressed using 3D equivalents of the 2D techniques used in still image compression. This philosophy captures temporal redundancy as well as spatial redundancy.

In this discourse 3D-SPIHT and 3D-ESCOT will be examined, as well as a recent memory efficient development.

The block diagram for a 3D transform system with ME/MC is presented below.

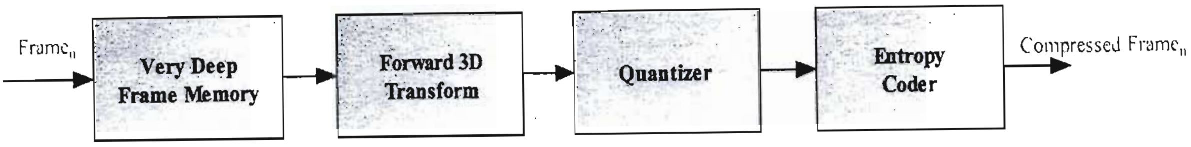


Figure 4-7: 3D Transform w/o ME/MC Block Diagram

Including motion compensation, as some of the systems to be presented do, yields the following diagram:

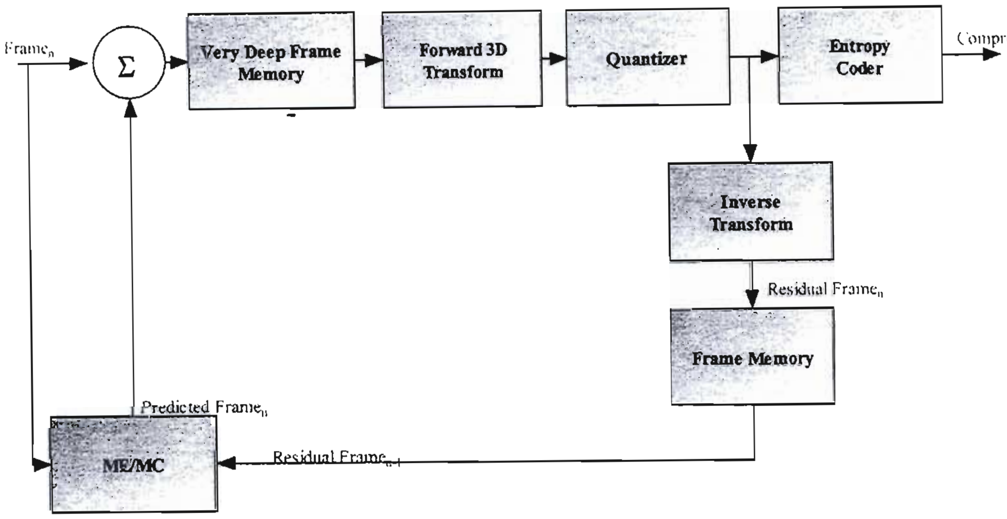


Figure 4-8: 3D Transform with ME/MC Block Diagram

4.2.2 Intra-frame Coders

The discussion of Section 4.2.1 has constructed a framework within which the literature methods may be considered. The following sections will present techniques and methods from the literature illustrating each one of the categories described above.

The simplest concept employed for coding a video sequence is to consider each frame as an independent still image. These individual images are then encoded using a known wavelet still image compression algorithm, such as those discussed in Chapter 3. As no advantage is taken of the temporal correlation between images, these techniques are expected to perform poorly.

4.2.2.1 Results

The literature review did not reveal any techniques reliant on intraframe coding. However, the algorithm proposed in Chapters 5 through 7 is based on intraframe coding. The results are given there.

4.2.2.2 Discussion

The inferior performance of this technique is clearly evident from the results. However, in certain applications, such as embedded processors where memory and computational resources are severely constrained, the intraframe coder may be the only viable solution.

4.2.3 Spatial Domain ME/MC

4.2.3.1 Introduction

The current generation of commercial video codecs is based on the standards described in Section 4.1. All these utilize spatial domain ME/MC and DCT coding. This is because the spatial domain method of block matching integrates well with the DCT framework already in place from earlier work in image compression, notably the JPEG algorithm. Several techniques have been proposed that incorporate wavelet techniques within this existing framework, thus leveraging current technology with new methods.

When considering motion estimation, there are two sources of motion within a scene; camera motion which includes panning, tilting and zooming, and object motion that includes rotation, translation and occlusion. The first source of motion causes the entire frame to be altered, thus techniques intending to overcome this effect are referred to as global motion compensation techniques. The second source of motion affects only certain areas of the frame, thus is dealt with by either block matching or object extraction techniques. All of the above ideas are furthered below.

4.2.3.1.1 Global ME/MC

Global motion compensation is a technique that compensates for effects such as camera panning, tilting or zooming. That is, the coordinates of the frame under scrutiny change, hence the term global. It is thus unsuited to describe motion within the frame, such as by rigid bodies. The best results that utilise global ME/MC also include 3D coding, and due to space constraints only these will be presented. Section 4.2.3.2 contains this discussion.

4.2.3.1.2 Local ME/MC

In existing techniques, such as MPEG and H.263, temporal decorrelation is achieved through motion estimation and compensation. Significantly, these motion functions are computed in the spatial domain. Several wavelet techniques have been proposed that take advantage of this; the wavelet transform achieves spatial decorrelation, while existing ideas of MC/ME achieve temporal decorrelation.

As Figure 4-3 above shows, the forward loop encodes a given frame using a wavelet transform. The feedback loop then performs an inverse transform, thus restoring the data to the spatial domain, in which the motion estimation and compensation are performed. The advantage of this is that proven techniques may be imported from existing DCT based algorithms, and implemented in conjunction with the wavelet transform. The disadvantage is that current techniques rely on block matching motion estimation, which is known to cause blocking artefacts; this will negatively affect the system performance. However, it is shown below that having motion compensation, even with this flaw, outperforms many schemes without compensation.

Several techniques consider ME/MC in conjunction with 3D coding Section 4.2.6 should be consulted for a discussion of such methods.

4.2.3.2 Block ME/MC – Shen and Delp

A technique that includes spatial domain ME/MC for local motion, and wavelet coding of the resulting residual frames, is presented by Shen and Delp in [Shen99]. They call this algorithm SAMCoW.

4.2.3.2.1 Premise

The algorithm operates under the assumption that wavelet still images coders have outperformed DCT based image coders, thus, by replacing the DCT coding stage of an MPEG algorithm, greater coding performance may be gained.

4.2.3.2.2 Algorithm

The algorithm operates by performing MPEG-like spatial domain wavelet coding. A GOF is formed of 100 to 150 frames and coded using intraframe and forward predictive motion compensated frames. The motion compensation is performed on 16x16 macroblocks and a motion vector search range of ± 15 pixels is used. This arrangement is similar to MPEG-1 but without the backward motion estimation mode. Each residual frame is processed with an overlapping block matching window to reduce the blocking effect. The process is conceptually similar to that described in the PACC coder (Section 4.2.3.4).

After the local ME/MC stage, each residual frame is encoded using a modified EZW coder, which exploits correlation between the colour planes, or chromatic redundancies. As this discussion has only concerned greyscale images to this point, this modification is not explored. For coding greyscale images, the coding is almost identical to the EZW still image coder.

An interesting, if not directly relevant, detail of this algorithm is the manner in which the coder handles reference frames during ME/MC. The EZW (Section 3.3.3) residual frame coder produces an embedded bit stream that is exploited to produce a rate scalable output stream. This has many advantages that have been explored. The difficulty which arises as a result of rate scalability, is that the reference frame at the encoder and decoder are different, at different bitrates. This can lead to the decoder producing erroneous output frames. For this reason, in this algorithm, the reference frames at the encoder and decoder are produced using the lowest bitrate available.

As indicated, this algorithm may be considered to be an MPEG-1 coder with the DCT residual frame coder replaced by the EZW coder.

4.2.3.2.3 Results

At high rates (1 – 6 Mbps) the following results were generated for the 'heavy' local motion sequences 'football' and 'flowergarden.'

	PSNR (dB)			
	Football		Flowergarden	
Rate	SAMCoW	MPEG-1	SAMCoW	MPEG-1
1 Mbps	27.1	28.8	24.3	25.7
2 Mbps	28.8	31.2	26.7	28.8
4 Mbps	30.9	34.2	28.7	32.6
6 Mbps	34.9	36.8	33.8	35.8

Table 4.1: SAMCow High Rate Results [Shen99]

Clearly the MPEG-1 outperforms the SAMCoW algorithm. At lower bitrates, the algorithm is compared to H.263 (the TMN number is not given in the paper) on the local motion sequence 'Akiyo' and the global motion sequence 'Foreman.'

	PSNR (dB)			
	Akiyo		Foreman	
Rate (kbps)	SAMCoW	MPEG-1	SAMCoW	MPEG-1
20	32.8	37.5	28.6	30.3
32	34.6	39.1	30.1	31.1
128	38.3	43.1	31.6	33.9
256	47.5	49.1	35.2	38.4

Table 4.2: SAMCoW Low Rate Results [Shen99]

Again the SAMCoW algorithm is outperformed by the DCT based H.263 algorithm.

4.2.3.2.4 Discussion

Although the results are good in terms of previous wavelet video coders, the most useful conclusion that may be drawn from this algorithm is that the zerotree coding algorithm is not as effective as the DCT in coding difference frames. This is possibly due to the previously mentioned effect of isolated high frequency coefficients forming, in violation of the zerotree coding model. A second explanation is, that applying the EZW routine to small tiles such as the 16x16 macroblocks used in MPEG, prevents the formation of large zerotrees, thus dramatically impacting on the performance of the algorithm.

4.2.3.3 Block ME/MC – Lin and Gray

A conceptually simple technique is proposed by Lin and Gray in [Lin01]. The authors propose encoding the residual frame produced by a ME/MC scheme using the SPIHT algorithm.

They note that the image model employed by zerotrees does not hold fully for residual frames. In particular in high motion scenes, each frame will contain significant high frequency coefficients, whose parent coefficients are not significant. This is in violation of the zerotree principle as outlined in Chapter 3. A large number of significance bits is required to describe the many insignificant parent coefficients of such an isolated significant coefficient. These bits do not contribute to a decrease in distortion, and are thus an overhead of the system. This is a similar argument to the one proposed by the same authors in [Lin02], Section 3.3.5., for an RD optimal SPIHT based still image coder.

They propose a solution based on the observation that significant coefficients tend to form in clusters in the wavelet domain. Thus the scheme calls for the image to be broken into blocks (16x16 in the implementation). Each of these blocks is then separately encoded using the SPIHT algorithm. The premise is that these clusters of significant high frequency coefficients cause the coding inefficiency; thus, by blocking the image, this inefficiency is limited only to a certain number of blocks, rather than the entire image.

Describing the image in terms of blocks allows bit allocation between the blocks to be performed. Thus the inefficient blocks may be assigned relatively more bits than the efficient blocks, to minimize the overall distortion. Lin and Gray adopt a Lagrange approach, almost identical to that described in Chapter 2. They also propose and interframe optimization based on the observation that due to the ME/MC the frames are dependant on one another. This process is not explored here.

The results from this system are very similar to H.263 as shown in Figure 4-9.

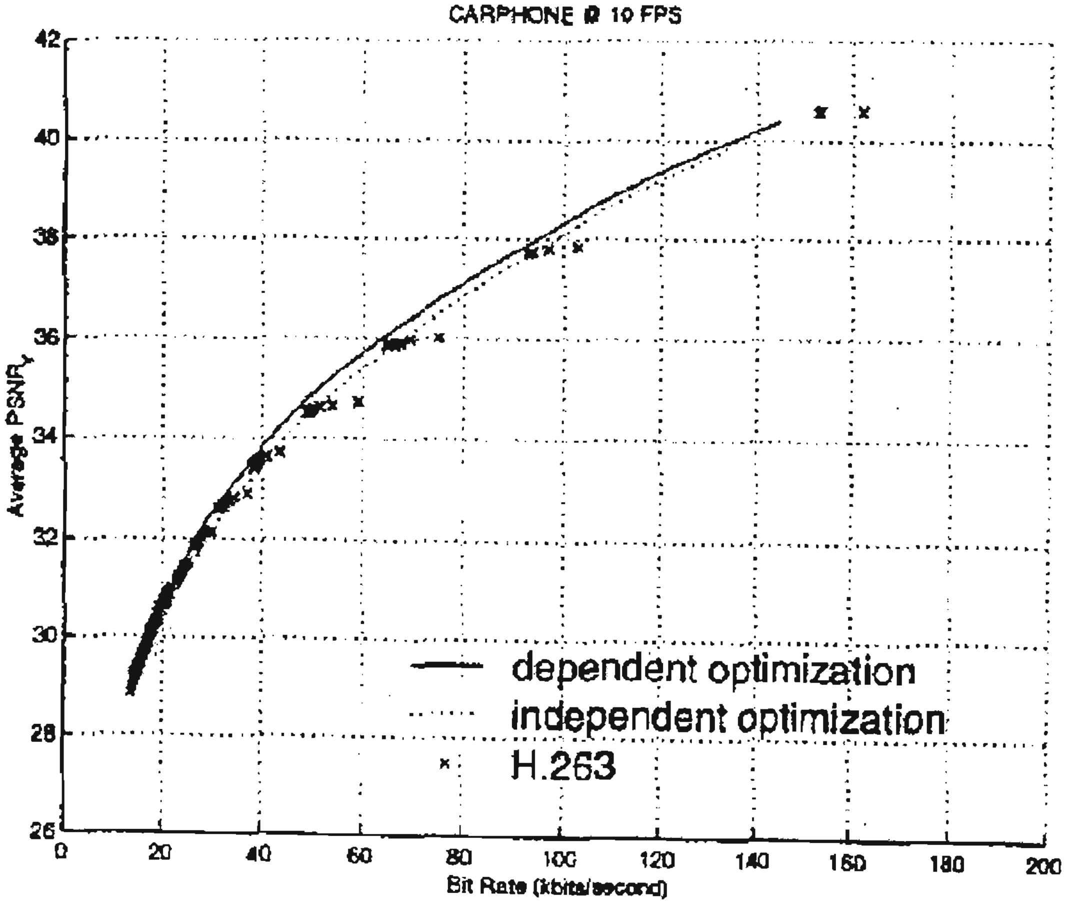


Figure 4-9: Results [Lin01]

4.2.3.4 Block ME/MC – Marpe and Cycon

Marpe and Cycon present an algorithm [Marp99] that uses block motion compensation followed by subband coding type wavelet transform scheme. The subband coding is based on context adaptive arithmetic coding. The block ME/MC and coding will be explored separately.

4.2.3.4.1 Block ME/MC

The basic block matching algorithm is very similar to that employed in the H.263 and MPEG algorithms. In addition, they employ overlapping block ME/MC (OBME/MC), as in H.263. This is important for their particular implementation.

The blocking artifacts introduced by block ME/MC are well known. The PACC algorithm first performs block ME/MC, and then wavelet transforms the entire frame. The edges caused by blocking will introduce many isolated high frequency coefficients in the wavelet domain, that the wavelet quantiser will be unable to code efficiently. These edges are an undesirable visual effect, and in addition reduce the efficiency of the wavelet quantization. The concept of overlapping blocks is to multiply groups of blocks by a window function, to reduce the visual impact of discontinuity across the block boundaries. Further details are to be found in the paper. This reduces the edge effect between blocks, and increases the efficiency of the frame based wavelet quantiser.

4.2.3.4.2 Coding

This paper introduces a new approach to wavelet coding that is most effective. It is referred to as partitioning, aggregation and conditional coding (PACC). It is in concept very similar to the ECECOW still image compression scheme described in Section 3.3.9.

After the ME/MC and wavelet transform, the wavelet domain is described by three datasets, the significance map, which describes the location of significant coefficients, the sign map that describes the signs of these coefficients, and the magnitude map, which describes the magnitude of the significant coefficients. This is the same data portioning as used in SPIHT (Section 3.3.4), but in this case each map is separate, and not in an embedded bit stream. This partitioning is performed due to two theorems given in [Marp99] which show that the entropy of a source is reduced by partitioning the source into separate subsources. This holds as each subsource will have different statistics, thus the variance of each one taken separately, is less than the variance of all the subsources taken together as one source. This is shown explicitly in Chapter 5. Each of the maps described above is such a subsource. This is the partitioning stage.

Aggregation refers to zerotree coding of each source, separately, in the usual manner.

Conditional coding refers to the description of each coefficient through a context. This context is formed by the sign and magnitude of a coefficient's neighbours. This stage is very similar to that employed in EBCOT (Section 3.3.12) and ECECOW (Section 3.3.9). This context is then used to drive an arithmetic encoder.

4.2.3.4.3 Results

The results of this coding technique are most impressive, outperforming MPEG-4 VM5.1. The following curves show the average PSNR plotted against the bit rate for the Akiyo and Hallmonitor Sequences.

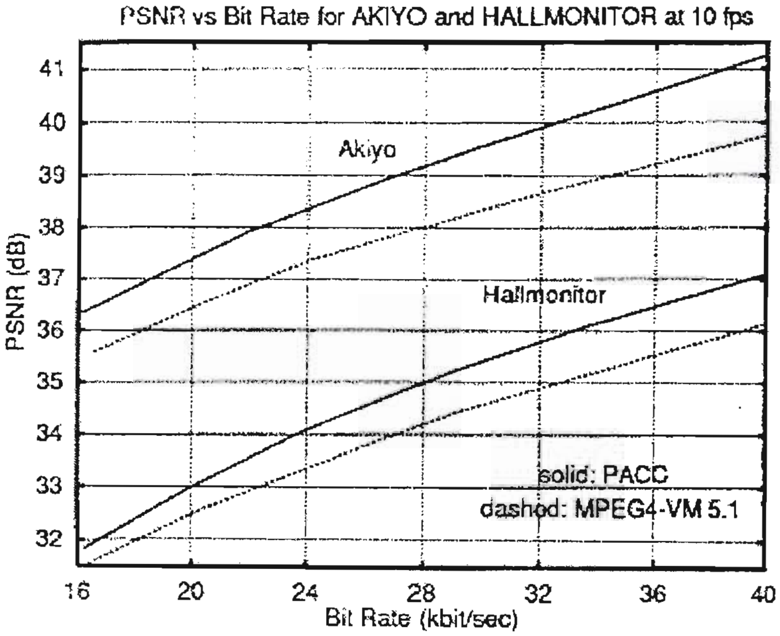


Figure 4-10: PACC Coding Results |Marp99|

Further results show the performance on a frame by frame basis.

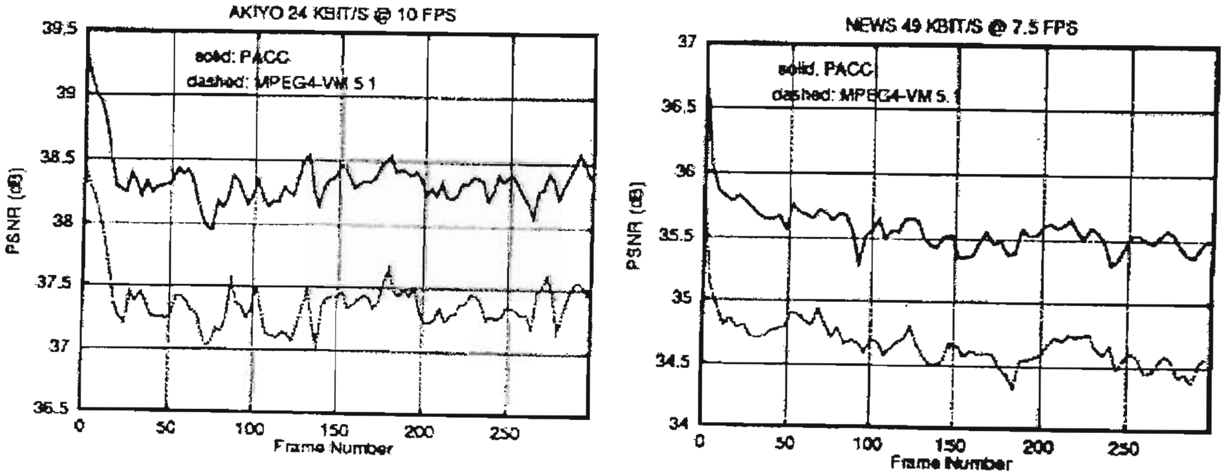


Figure 4-11: Further Results for PACC |Marp99|

4.2.4 Wavelet Domain ME/MC

There is a school of thought that asserts that advantage is to be gained by providing the ME/MC directly in the wavelet domain. This school cites both the advantage gained by not having first to spatially encode and then decode the image, as in standard ME/MC techniques, as well as hoping to discover that the wavelet domain exposes certain video characteristics that may be exploited.

This field is large and beyond the scope of this document, thus the discussion will be limited to a single representative algorithm [Yang00] which proposes a complete and highly successful video coding algorithm with ME/MC in the wavelet domain. The major problems with wavelet domain ME/MC is discussed during the course of the development. The following development is taken from [Yang00].

4.2.4.1 Aliasing in the Wavelet Domain

The major problem in performing motion estimation in the wavelet domain arises from the aliasing present in this domain. The wavelet transform may be performed by the following filter bank arrangement, (see Appendix A for details):

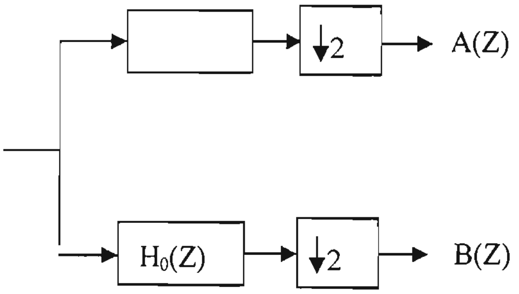


Figure 4-12 : Single Level Wavelet Decomposition

The input to the downsampling block is the convolution of X(Z) and $H_0(Z)$, which in the Z-domain is simply their product, $F(Z) = X(Z)H_0(Z)$.

The downsampling operation is described in the Z-domain by the following:

$$A(Z) = \frac{F\left(\sqrt{Z}\right) + F\left(-\sqrt{Z}\right)}{2}, \tag{4.1}$$

where the term in negative Z represents aliasing. This is clearly seen by transforming into the ω domain:

$$A(\omega) = \frac{1}{2}\left(F\left(\frac{\omega}{2}\right) + F\left(\frac{\omega}{2} + \pi\right)\right). \tag{4.2}$$

This aliasing interferes with almost all signal processing in the wavelet domain. For instance, in traditional motion compensation, the translation of groups of pixels is calculated. Translation in the spatial domain results in a linear phase shift in the frequency domain:

$$(Ff(x-n))(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\omega(x-n)} f(x-n)dx = e^{i\omega n}(Ff)(\omega). \tag{4.3}$$

Thus, the Fourier transform is shift invariant to within a phase shift, however as (4.2) demonstrates, the wavelet domain is not. This problem has complicated attempts at motion compensation in the wavelet domain.

4.2.4.2 Optimal Alias Reduction

In their paper [Yang00], Yang and Ramchandran propose to alleviate the aliasing problem through the use of optimal filtering. With the goal of balancing the removal of aliasing energy with the preservation of signal energy they formulate a Lagrangian optimization, $L = S + \mu T$, where S is the MSE error of the filtered output, and T is the aliasing energy. Through this they arrive at an optimal filter design.

In Figure 4-12, consider a signal $x[n]$ then the filtered and downsampled signal $A(\omega)$ is given by (4.2). Now a linear translate, $y[n] = x[n-v]$, of the original signal will have a frequency domain representation calculated through (4.3) as:

$$Y(\omega) = \tfrac{1}{2}\left(F\left(\tfrac{\omega}{2}\right)e^{-j\frac{\omega}{2}v} + F\left(\tfrac{\omega}{2}+\pi\right)e^{-j\left(\frac{\omega}{2}+\pi\right)v}\right). \tag{4.4}$$

The original filtered signal has a spectrum given by:

$$X(\omega) = \tfrac{1}{2}\left(F\left(\tfrac{\omega}{2}\right) + F\left(\tfrac{\omega}{2}+\pi\right)\right) \tag{4.5}$$

As is evident, both translation and downsampling introduce a phase distortion into the signal spectrum. The interaction between these two effects gives rise to the situation in (4.4) where the aliased (second) term has a different phase shift to the unaliased term. Thus the effect of aliasing cannot be removed simply by subtracting (4.4) from (4.5).

Yang and Ramchandran suggest the following scheme, where both the original and translated signals are upsampled and filtered by a filter, $L(\omega)$, prior to motion estimation.

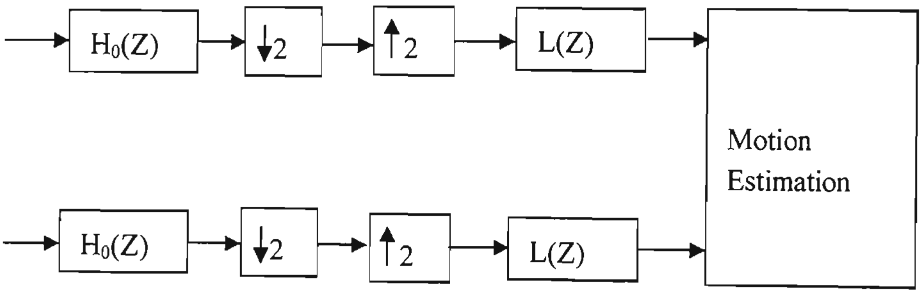


Figure 4-13: Filtering for Motion Estimation the Wavelet Domain

In order to ameliorate the aliasing effect through filtering, the filter must remove as much of the aliasing spectral energy as possible, while maintaining the signal energy. The original signal, after the final filter may be written as:

$$X_2(\omega) = \tfrac{1}{2}\left(F\left(\tfrac{\omega}{2}\right) + F\left(\tfrac{\omega}{2}+\pi\right)\right)L(\omega). \tag{4.6}$$

Preserving the signal energy requires that the first term in (4.6) equals $X(\omega)$, thus the optimization critertia S may be written as:

$$S = \int_{-\pi}^{\pi} \left|\left(X(\omega) - \tfrac{1}{2}F(\omega)\right)\right|^2 d\omega = \int_{-\pi}^{\pi} \left|\left(X(\omega)\left(1 - \tfrac{1}{2}L(\omega)\mathrm{H}_0(\omega)\right)\right)\right|^2 d\omega \quad , \tag{4.7}$$

where the error is stated in the mean square sense.

The second criteria is that the aliasing signal energy must be minimized, which requires that the second term in (4.6) tends towards zero.

$$T = \int_{-\pi}^{\pi} \left|H_0(\omega+\pi)L(\omega+\pi)G(\omega)\right|^2 d\omega \tag{4.8}$$

Thus forming a Lagrange minimization problem

$$F(L;\mu) = S + \mu L = \int_{-\pi}^{\pi} \left|1 - \tfrac{1}{2}H_0(\omega)L(\omega)\right|^2 + \mu \cdot \left|H_0(\omega+\pi)L(\omega+\pi)\right|^2 \left|X(\omega)\right|^2 d\omega \quad . \tag{4.9}$$

The Lagrange .multiplier may now be found using the methods described in Chapter 2. However, it still remains to solve the filter coefficients L. Considering the operation in the time domain, that is, writing the filtering operations as convolutions, taking the derivative with respect to L and equating to zero to find the minima, yields the following form:

$$l = \left[\mathbf{C}^{-1}\mathbf{R}_{\mathrm{x}}\mathbf{C} + \mu\mathbf{C}_{\phi}^{-1}\mathbf{R}_{\mathrm{x}\phi}\mathbf{C}_{\phi}\right]^{-1} \cdot 2\mathbf{C}^{-1}\mathbf{R}_{\mathrm{x}}\delta , \tag{4.10}$$

where $\mathbf{C}$ is the convolution matrix of the filter $H_0(\omega)$, and $\mathbf{R}_{\mathrm{x}}$ is the correlation matrix of $x[n]$, finally the subscripted ϕ refers to the quantities with respect to the complimentary band filter, $(-1)^n h_0[n]$.

Knowing the filter $H_0(\omega)$, for instance the popular Daubechies 9-7 filter, and making an appropriate statistical model of the input signal, and optimal alias filter coefficients set, l, may be derived.

Applying this filter to the signals prior to motion compensation will reduce the aliasing noise, thus allowing for a better motion estimation.

4.2.4.3 Hierarchical Motion Estimation

Hierarchical motion estimation refers to the family of techniques that perform motion estimation through a multiresolution analysis. Motion estimation at a coarse scale is used as the first approximation for motion at the next finer level of resolution. Naturally the wavelet transform is ideally suited to such an analysis.

[Yang00] presents the following method. The following notation is used to simplify the discussion, LL_t^n, where LL refers to the subband, either LL, HL, LH or HH, at resolution level n, and frame at time t.

Two consecutive frames are decomposed with the wavelet transform, to N resolution levels. Starting at n=N the following process is performed.

- Apply the alias reducing filter to the reconstructed LL_{t-1}^n and LL_t^n bands.

- Perform block matching ME on these image subbands. Both the encoder and decoder perform the ME, thus the motion vectors do not need to be transmitted. Thus a dense motion field, using 4x4 macroblocks is required.

- Reconstruct $\left(LL_{t-1}^n, LH_{t-1}^n, HL_{t-1}^n, HH_{t-1}^n\right)$ to obtain LL_{t-1}^{n-1}, the LL band at next finer resolution. Apply the motion vectors from the previous step to this image to obtain the MC estimate LL_t^{n-1}, of LL_t^{n-1}. Apply an overlapping block matching window, to reduce the blocking effect, as in PACC (Section 4.2.3.4).

- Decompose $\hat{LL}_t^{n-1}$ to obtain the MC estimate $\left(\hat{LL}_t^n, \hat{LH}_t^n, \hat{HL}_t^n, \hat{HH}_t^n\right)$, of $\left(LL_t^n, LH_t^n, HL_t^n, HH_t^n\right)$.

- Find the residual bands $\left(LH_t^n - \hat{LH}_t^n, HL_t^n - \hat{HL}_t^n, HH_t^n - \hat{HH}_t^n\right)$, and code these bands using the EQ coder described in Chapter 3, Section 3.3.10. This coder is chosen because modelling the MC wavelet subbands as a mixture process of GGD with slowly changing variance is accurate for coding residual frames. In addition, the EQ coder provides the flexibility to code each subband independently with different statistics, as described in PACC; such a source partitioning arrangement has performance advantages.

- Synthesise $\left(\hat{LL}_t^n, \hat{LH}_t^n, \hat{HL}_t^n, \hat{HH}_t^n\right)$ to form the estimated subband $\hat{LL}_t^{n+1}$, and repeat the process, until the entire wavelet multiresolution pyramid has been processed.

As can be seen the motion estimate at one resolution level is used as the initial estimate at the next level, hence the term hierarchical motion estimation.

There are further coding considerations in this algorithm, such as an optional forward motion estimation scheme, but these are sundry to our discussion and are omitted.

4.2.4.4 Results

The performance of this algorithm at low bit rates was compared to H.263 on the 'Mother and Daughter' and 'Miss America' sequences, both are local motion sequences. The following characterizes the performance.

Sequence	Bit rate (kbps)	Frame Rate (fps)	H.263 (dB)	G (dB)	L (dB)
M & D	48	15	34.8	35.1	35.6
Miss America	14	15	36.3	36.6	36.9

Table 4.3: Low Rate System Performance [Yang00]

In the table above, G refers to the described scheme without the use of the optimal alias reduction filtering, and L refers to the full scheme. It is clear that the alias reducing scheme has performance merits.

The high rate performance is compared against MPEG-2 for the Football sequence, a high motion sequence.

Bit Rate (Mbps)	MPEG-2 – PSNR (dB)	L – PSNR (dB)
0.5	25.1	27.26
2	30.0	31.6
7	37.1	37.7
10	40.6	41.0

Table 4.4: High Rate System Performance [Yang00]

It is evident that at high rates the system still has favourable coding performance.

The above results show that this proposed scheme has good low and high rate performance.

4.2.4.5 Discussion

The major advantage of this scheme is the backward motion estimation. It may be seen that the high rate performance is better than MPEG by a larger margin, at the lower bit rates. This is because the rate contribution of the motion vectors in the MPEG-2 system becomes significant at these rates. Due to the backwards motion estimation employed by this scheme, motion vectors do not need to be transmitted, this is a major advantage.

The disadvantage is that ME/MC needs to be performed at the decoder as well as encoder with this scheme, whereas MPEG-2 and H.263 only require ME/MC at the encoder. As ME/MC is a computationally expensive process, this adds significantly to the complexity of the decoder.

In addition, the algorithm demonstrates the need for alias reduction when performing motion related processing in the wavelet domain.

4.2.5 3D Subband Wavelet Coding

3D Subband coding is a major competitor against motion estimation based techniques. It has the advantage that motion estimation schemes rely on exhaustive block matching searches, which are computationally expensive. This idea is explored further in Chapter 5. 3D Coding is far less computationally demanding, however is demanding in terms of memory.

The underlying idea is that the time axis may be handled in the same way as the spatial axes. Correlation structures should exist in the time direction and the wavelet transform should be able to provide a good representation for these redundancies.

An important feature of theses 3D wavelet coders is that the bitstream may be constructed in an embedded fashion, like the 2D wavelet coders. This allows the spatial resolution, frame rate, and bit rate all to be finely controlled. This is becoming an important feature in deploying video over computer networks that may be intolerant to bursty traffic. Usually motion estimation based video coders are not able to provide this functionality.

Two leading schemes will be presented below, which do not employ motion estimation, and thereafter schemes that do incorporate motion estimation will be explored.

4.2.5.1 3D SPIHT

Based on the success of the SPIHT algorithm in still image coding, Pearlman and his collaborators have proposed and continued developing a 3D version of the algorithm; 3D-SPIHT. The original concept was proposed in [Kim97] and has evolved several times to the version discussed here, presented in [Kim00].

4.2.5.1.1 Premise

It is argued that as there is high interframe correlation, if a pixel is insignificant in one frame, it will be so in the subsequent frames too. Thus the 2D concept of a zerotree may be extended to include the temporal direction, to encode this correlation.

4.2.5.1.2 Algorithm

The 3D-SPIHT coder splits the video sequences into groups of frames (GOFs) of typically 16 or 32 frames, which are considered separately. Each frame within the GOF is independently wavelet transformed. Kim et al propose a 3D zerotree structure, as illustrated below.

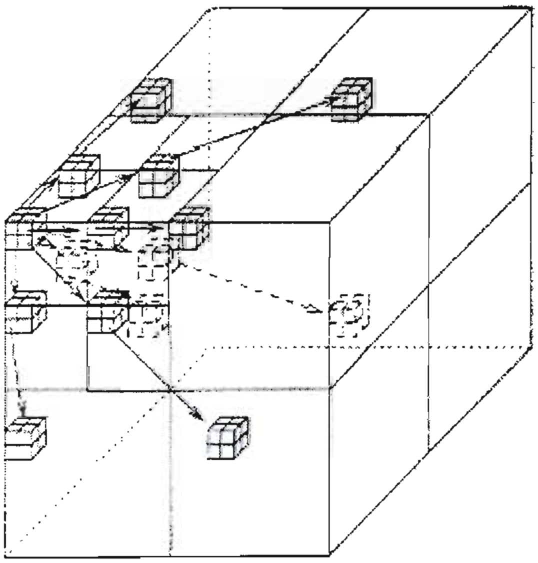


Figure 4-14: 3D Zerotree Structure [Kim00]

The algorithm traverses the 3D subband structure in a scanning order similar to SPIHT, outputting bits according to the same rules as SPIHT, but aware of the third dimension. Details of the construction of the bitstream are not relevant to this discussion, and are omitted, except to note that they are similar to the 2D case.

4.2.5.2 3D ESCOT

As with the SPIHT algorithm, the EBCOT (Section 3.3.12) algorithm may be extended to account for temporal correlation. The algorithm is proposed in [Xu01], and extended in [Xu02]. As with EBCOT the core of the algorithm is a context based adaptive arithmetic coder.

4.2.5.2.1 Premise

The premise is the same as for EBCOT. Each subband has unique statistical properties that relate the probability of a coefficient being significant at a bitplane level, to the significance of the coefficient's neighbours. By forming a context for a coefficient based on its neighbours, an arithmetic encoder can efficient encode each subband.

In addition, traversing each subband in bitplane order allows the algorithm to create an embedded stream for each subband. These streams may then be truncated to achieve a global RD optimization.

4.2.5.2.2 Algorithm

The video sequence is split into GOFs of 32 frames. Each GOF is considered separately, and each frame in the GOF is wavelet transformed with a usual 2D transform. The resulting coefficient frames are then wavelet transformed in the temporal direction.

Each subband created by this process is then considered independently. This is slightly different to EBCOT, which breaks each subband into tiles for consideration. The subband is traversed in bitplane order and a context is formed for each coefficient based on the value of its neighbours, both in space and time. The process is much like EBCOT and is not expanded here. The resulting stream is encoded using an arithmetic coder. The output stream is embedded and may be truncated at any point.

A global RD optimization is performed, by truncating each subband output stream so as to minimize the overall distortion. This again mirrors the EBCOT procedure.

4.2.5.2.3 Results

The results from this algorithm are most impressive, as shown in the table below.

	PSNR (dB)					
	Akiyo		Mother & Daughter		Coast Guard	
Rate(kbps)	20	40	20	40	40	80
3D ESCOT	31.32	35.55	31.91	34.90	26.61	28.39
3D SPIHT	29.43	33.38	30.09	33.33	25.88	27.65

Table 4.6: Comparison between 3D SPIHT and 3D ESCOT [Xu01]

The same sequence is illustrated on a frame by frame basis for the Mother and Daughter sequence at 20 kbps:

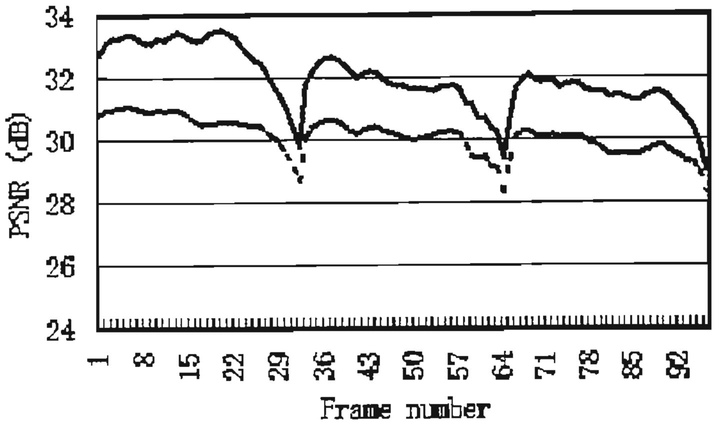


Figure 4-16: 3D ESCOT Performance Comparison [Xu01]

As with the 2D case, 3D-ESCOT outperforms 3D-SPIHT due to its context model which is superior to the zerotree model.

4.2.5.2.4 Update

A limitation of the 3D coders presented above is the manner in which the GOF structure is handled. The sequence of frames is split into GOFs that are handled independently. When considering frames toward the end of the GOF the wavelet transform has no future frames to consider, and will thus introduce a boundary effect. This is manifested as a severe drop in the PSNR at frame boundaries, as illustrated in Figure 4-16 above.

Xu et al present a solution in [Xu02]. Previous schemes split the video sequence into GOFs initially. This algorithm considers the GOF centred around each frame continuously. A buffer is created that contains a GOF, ie 16 frames. Each frame is wavelet transformed spatially and then the group is transformed temporally, as before. However, once the 3D ESCOT algorithm has been run for the first frame, it is removed from the buffer, conceptually each frame is then shifted forward one, and another frame inserted in the rear of the buffer. The process is repeated for each frame. In this way a quasi-infinite buffer is created.

Results from this scheme are very impressive, outperforming MPEG-4 (Microsoft Version 17.0, which is available to these authors due to their affiliation with Microsoft) significantly. For instance, the 30 fps Akiyo sequence, encoded at 42 kbps, produces the following results.

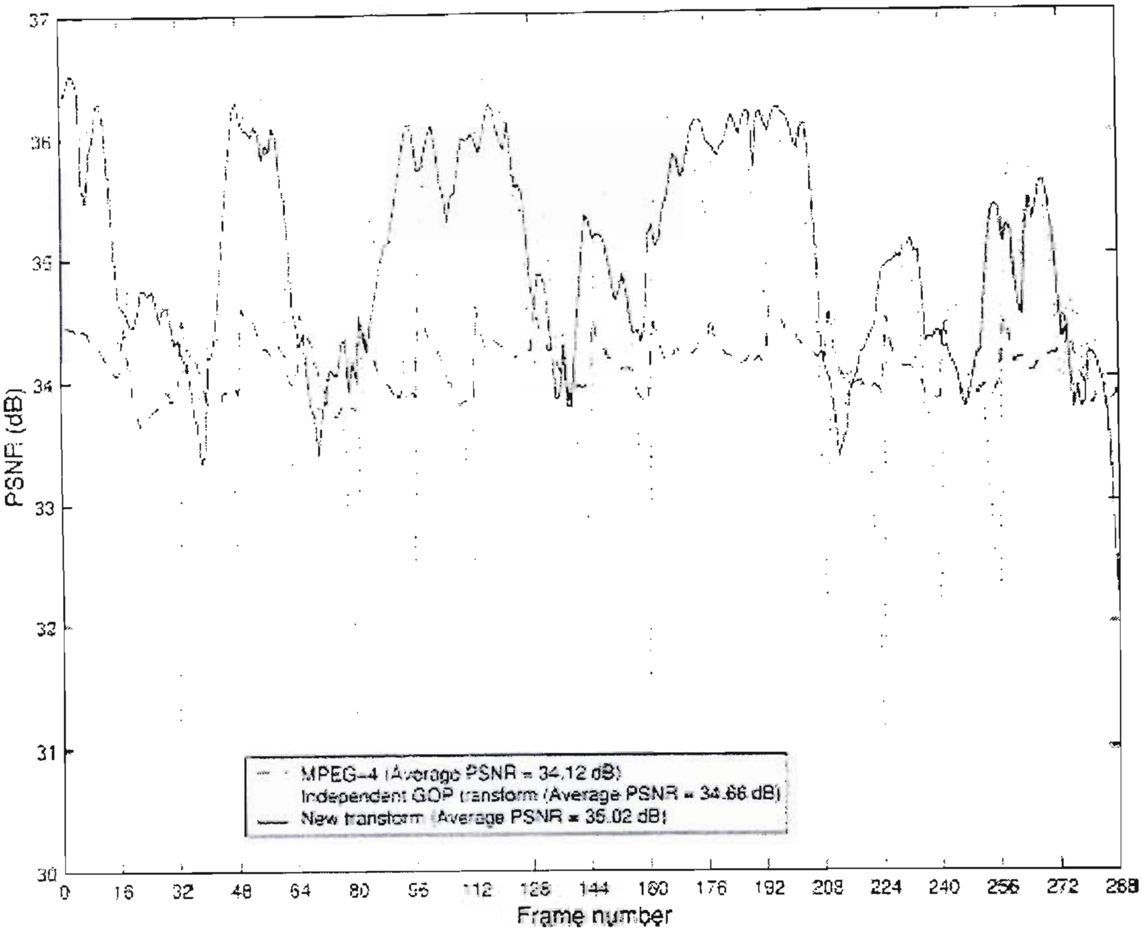


Figure 4-17: 3D-ESCOT Update Results |Xu02|

The PSNR dipping in the "independent GOP" transform is clearly apparent, and the removal of this effect is equal clear with the new codec.

4.2.6 3D Subband Coding with ME/MC

Early research into 3D subband coding produced results that were inferior to ME/MC based techniques [Wang02]. The reason is that the ME/MC techniques are very successful in capturing motion, and thus produce an error frame for coding that has very low entropy and may be efficiently compressed. Stated alternatively, the model used by ME/MC coders accounts for motion, which standard 3D coders do not, thus as the model more accurately reflects the source, better compression results.

Due to this shortfall there has been much work in combining motion estimation techniques with 3D subband coding. The following two sections will give a brief overview of the approach and results obtained by two of these methods.

4.2.6.1 Global ME/MC

In cases where the camera moves or zooms, every pixel in the frame will be affected. Describing this effect through block based ME/MC is inefficient, and other techniques have been discovered. This section will give an overview of one such technique that has been applied in conjunction with 3D-SPIHT.

Wang et al describe a technique in [Wang02] that handles the global ME/MC as a preprocessing step before applying 3D-SPIHT. The global ME/MC is achieved through affine transform.

A GOF is formed, and for each frame except the first, a block matching process is performed to calculate the alignment of the frame with respect to the first frame. Using a numerical process, such as a least mean squares allows this alignment to be expressed in terms of an affine transform. Such a transform has six parameters that represent geometrical warping; rotations, shifts and zooms.

Each image is then transformed using the affine transform, effectively aligning it with the first frame; the parameters of these transforms are sent as side information. The group of warped frames is then compressed using 3D-SPIHT (Section 4.2.5.1). As each frame should be very similar to the previous, as a result of the affine transform, the temporal correlation should be high, and the compression effective.

Results were calculated for the "Coast Guard" sequence at 30 fps, which is a sequence dominated by a global panning motion. The following table shows the average PSNR with various algorithms.

	PSNR (dB)	
	50 kbps	100 kbps
H.263 (TMN9)	26.71	29.64
3D SPIHT	26.65	28.75
3D SPIHT with global ME/MC	27.21	29.75

Table 4.7: Global ME/MC 3D SPIHT PSNR Results [Wang02]

It is evident that there is a performance gain through the ME/MC process. However, it is noted that due to the large number of floating point operations during the affine transform process, the proposed algorithm is an order of magnitude slower than H.263. In addition the 3D SPIHT algorithm requires the entire GOF to be buffered in memory, which is often impossible. Thus this technique is effective, but outperformed in terms of rate-distortion, and complexity by H.263.

4.3 Discussion of Wavelet Video Compression and Conclusion

This chapter presents a thorough review of wavelet video compression. The current DCT and ME/MC based standards are listed, and their key features briefly explored. Thereafter, several proposed wavelet methods are presented, grouped by operating principle.

It is extremely difficult to compare these algorithms, as their published results are minimal and based on different test sequences at different bit and frame rates. However, it can be seen that the wavelet methods are competitive with the standard methods. However, the large performance gains that wavelet techniques have gained over DCT techniques in still image coding are not realised for video compression. This is on account of the difficulty in providing a temporal syntax in the wavelet domain. As discussed, blocking the image interferes with the formation of zerotrees due to edge effects, thus the traditional ME/MC methods are not absolutely suitable.

In addition, the relative novelty of wavelet video coding, and hence, the relatively smaller volume of research suggest that large gains may be made in the future. In contrast MPEG is very well established, and MPEG-4 is a most sophisticated algorithm.

Section 4.2.3 presents methods that impose a block structure onto the wavelet transform to achieve ME/MC in a similar fashion to MPEG coders. [Shen99] present a very simple algorithm that directly replaces the DCT algorithm in MPEG-1 with an EZW based algorithm. This algorithm subsequently performs worse than MPEG-1. [Lin01] presents results that are most similar to H.263, while [Marp99] exceeds the coding capabilities of MPEG-4, although the results are only given at high frame rates.

Section 4.2.4 presents ME/MC in the wavelet domain. Here [Yang00] present results that are superior to H.263 at low rates, and slightly greater than MPEG-2 at high rates.

Section 4.2.5 presents 3D subband coding, both with and without ME/MC. [Kim00] finds a 3D-SPIHT algorithm, either with or without ME/MC, to be inferior to H.263, although computationally simpler (if probably more memory intensive) [Xu01] and [Xu02] present a 3D coding routine without ME/MC than outperforms 3D SPIHT. Unfortunately no comparison to H.263 is given, although, based on our experiments in Chapter 7, the algorithm outperforms H.263. Furthermore the algorithm is found to be superior to MPEG-4 for a low and high motion scene. Finally [Wang02] presents a joint global motion estimation and 3D subband coding scheme that, at moderate bit rates, outperforms 3D SPIHT and H.263 for a high global motion scene.

Unfortunately no direct comparison between these algorithms is possible, as each paper has used a different configuration of test sequence, frame rate and bit rate. Several general statements are possible however.

- Local motion estimation through blocking presents possible RD performance advantages, however careful design of the scheme must be undertaken so as not to cripple the wavelet spatial quantisation stage. The success achieved in [Lin01] and lack thereof in [Shen99] is on account of such considerations.

- Motion estimation in the wavelet domain is both theoretically and practically challenging. The new concept of hierarchical motion estimation does offer the promise of future gains due to its direct integration with the wavelet transform representation.

- 3D subband coding presents varied success. In all cases this performance is at the expense of large memory requirements as well as high frame rates, which are often prohibiting factors. The success achieved in [Wang02] in aligning frames prior the 3D subband decomposition indicates that if good ME/MC can be achieved for local motion, similar gains should be possible.

At this point there is no clearly 'better' option. ME/MC in the spatial domain appears to be the computationally simplest means of temporal decorrelation, while presenting competitive results. However, wavelet domain ME/MC appears to have greater scope for advancement as it integrates more closely with the wavelet domain. 3D subband coding presents very good results currently, particularly for global ME/MC, but the large memory requirements suggest that this may not be a particularly elegant solution.

For the purposes of this project, computational simplicity is of primary importance. Thus [Lin01] and [Marp99] were primary references in the design of the new algorithm in Chapter 6.

Chapter 5 - Source Model for Proposed Algorithm

The previous chapters have undertaken a wide literature review to reveal many of the ideas and algorithms used in video compression. Chapter 2 introduces RD theory which is used to optimise the behaviour in many video coders. Chapter 3 examines still image coding algorithms as these algorithms are often used in video coders for spatial coding. Chapter 4 examines existing video coders, as well as the proposed wavelet methods. This chapter will proceed from this general theoretical basis, to design a video coding algorithm for the intended application of this project.

The goal of this project is to produce a video compression system for a mobile device. The intended platform is a digital signal processor (DSP), with limited external RAM. This platform imposes severe restrictions on the computational and memory complexity of the algorithm. In addition, the target device is connected to a low bandwidth radio channel. These factors must be considered in comparing the various techniques proposed in the preceding chapters.

The greatest concern is system complexity, thus the first stage in the design was to determine the complexity of the existing standards. Specifically the complexity of each coding stage was examined, to determine where the greatest complexity savings could be made.

Section 5.2 details this study, and reveals that the block ME/MC that is foundational in MPEG, H.263 as well as the most successful wavelet video coders presented in Chapter 4 is the primary contributor to the complexity of these algorithms. The extent of the complexity of ME/MC suggested that any attempt to implement an algorithm based on block ME/MC on general hardware would be unsuccessful. This observation is borne out by the lack of such software in the market. Thus in order to meet the project imperative of producing working video compression software for a DSP, block ME/MC was abandoned.

The problem that remains is to achieve temporal decorrelation. This chapter will explain how the decision to use difference frames for temporal decorrelation was arrived at. Having made this decision a suitable coding strategy had to be derived. Chapter 3 shows the primary importance of finding a good source model. Thus a nature of difference frames is examined in Section 5.3 in order to develop a model.

Thus this chapter is primarily concerned with the design of the temporal decorrelator for the proposed system.

5.1 Overview of Process

The first stage of the process was to determine the complexity of existing standard systems. To this end the reference software distributions for the H.263+ codec [H263], MPEG-2 codec [MPEG-2], and MPEG-4 [MPEG-4] were obtained. Several standard video sequences were processed using these codecs, and the Microsoft Profiler (an element of Microsoft Visual Studio 6.0) was used to obtain timing information, from which a complexity estimate may be made.

The striking outcome of this study, which is detailed in Section 5.2, is that motion estimation consumes between 80 and 90% of the computational time. Based on this result, the slightly radical decision was taken to attempt to perform video compression without motion-estimation and compensation (ME/MC), in order to produce a low complexity video coder.

The only other form of temporal decorrelation presented in the literature takes the form of 3D transforms. Unfortunately these schemes are also inappropriate for the system, owing to their memory requirements. For instance 3D ESCOT, (Section 4.3.5.2) requires a 16 frame buffer.

The only alternative is a system based on difference frames, as described in Section 4.3.1.2. This is the simplest form of temporal correlation, requiring only the subtraction of the previous frame from the current. The memory requirements are thus a double frame buffer, and the computational requirement is only 1 subtraction per pixel. A scheme based on difference frames will require significantly more modest memory and computational resources than any other scheme presented in the literature review.

Attempting to achieve temporal decorrelation using difference frames is unpopular. Taking the difference between two frames takes no cognisance of the underlying physical phenomena that give rise to temporal correlation, whereas ME/MC schemes naturally do.

However, this thesis proposes that previous arguments against using difference frames are flawed. Difference frames display strong spatial clustering of significant coefficients, which motivates for the use of a spatially adaptive bit allocation strategy for their encoding. The benefits of such a scheme are described. In this way the underlying model that produces difference frames is exploited. In addition these properties allow the idea of tiling and spatial bit allocation, presented for motion compensated frames in [Lin01] to be modified and utilised.

There is no argument that performing ME/MC will produce a residual frame with a lower variance than a similar frame produced by using difference coding alone. This will allow better RD performance upon compression, as shown in Chapter 2. However, in the case of low complexity coding, difference frames have computational advantages that may outweigh the RD performance cost. This chapter shows how the use of RD classification can improve the RD performance of difference frame based systems, thus enabling a very low computational and memory complexity video coder.

Chapter 6 following, presents the complete compression scheme based on the machinery proposed in this chapter. This scheme is then compared to other methods.

5.2 Complexity Study of Video Coding Standards

In order to produce a low complexity algorithm, a study of the complexity of existing codecs was undertaken. As the ITU and ISO publish reference software for their standard H.263+ and MPEG algorithms respectively, these algorithms are explored.

5.2.1 H.263

As the target system is intended for use over low bandwidth channels, most effort was spent on H.263 (Section 4.2.4). Of the standardised algorithms investigated, H.263 is designed for the scenario closest to that for which the new algorithm is intended, thus it is a very useful benchmark.

Owing to the size, and source dependence of H.263, developing an analytical complexity model is both cumbersome and probably inaccurate. For this reason empirical complexity data was gathered. ITU reference software [H263] was obtained and compiled. A range of standard test sequences was compressed at various bit rates. Each sequence is in QCIF format (176x144), subsampled at 4:2:0, with only the luminance information being non-zero. In each case the Microsoft Profiler (part of the Microsoft Visual Studio 6.0) was used to obtain timing information for each function.

Full results of this study are presented in Chapter 7, however a selected subset is presented below in Table 5.1

Sequence	Akiyo					Hallmonitor				
Rate (kbps)	15	20	25	40	50	15	20	25	40	50
Frame Rate (fps)	7.3	9.8	10.6	11.3	11.5	7.6	9.8	10.1	11.2	11.3
Total Time (ms)	26137	38667	38814	39128	39138	15596	38450	39543	39323	42120
ME/MC (%)	85.33	83.47	82.94	86.08	84.70	64.12	82.95	82.82	85.02	79.79
DCT (%)	0.56	0.59	0.63	0.63	0.55	0.62	0.61	0.56	0.56	0.57

Sequence	News
Rate (kbps)	20
Frame Rate (fps)	10.7
Total Time (ms)	39095
ME /MC (%)	87.32
DCT (%)	0.51

Table 5.1 H.263 Complexity Results

In each case the motion estimation and compensation accounts for approximately 80% of the computational burden.

Although the results presented above are specific to the environment in which they were generated (which is detailed in Chapter 7), the contribution of the ME/MC will remain constant across all platforms.

5.2.2 MPEG-2

Although MPEG-2 is unable to achieve the low bitrate coding targeted by our algorithm, the complexity of the system is of interest. For this reason an identical experiment to that described for H.263 was performed using the MPEG-2 algorithm. The MPEG-2 codec [MPEG-2] used is the reference software release from the ISO body.

Results are shown in Table 5.2 for the Akiyo sequence. MPEG-2 does not support variable frame rate, as it was designed for broadcast video. Thus it should be noted that all the sequences are encoded at 30 fps. Thus a direct comparison with the H.263 results may not be made from these results.

Sequence	Akiyo		
Rate (kbps)	40	60	100
Frame Rate (fps)	30	30	30
Total Time (ms)	25112	25574	25239
ME /MC (%)	40.33	40.11	40.33
DCT (%)	26.5	26.0	26.4

Table 5.2: MPEG-2 Akiyo Complexity Results

The complexity contribution of the ME/MC is again significant, if less so than the H.263 case.

As expected from the discussion of H.263 and MPEG-2 in the Chapter 4, MPEG-2 encodes more data in less time; thus may be seen as more computationally efficient. This is because it is a simpler algorithm; in particular the ME/MC is simpler in MPEG-2, as expressed in the Table 5.2. This computational advantage has a resulting RD performance disadvantage, as will be shown in Chapter 7.

5.2.3 MPEG-4

In order to compare the system to MPEG-4, the official ISO reference software [MPEG-4] from ISO was obtained. There are two versions of the reference software.

The first version investigated is produced by the Microsoft Corporation. Unfortunately the rate control aspects of this software do not function, as stated in the User Manual [MPEG-4], "Rate control is currently broken for both MPEG-4 and TM5 style." Thus, although one may obtain executable codecs with rate control, the source code version of this software is unobtainable. As this aspect is crucial in obtaining streams at different rates, this is a serious limitation.

The second version, MoMuSys, is designed for use with unix-like utilities, and converting it for use with Microsoft Visual C++ in order to perform a complexity analysis was considered beyond the scope of this work.

Thus only a reduced complexity analysis was performed. The standard test videos were encoded with the Microsoft MPEG4 reference code, without rate control. It should be noted that only the block based coding capabilities of MPEG-4 were employed, not the shape based coding. The following results were obtained.

Sequence	Akiyo	News	HallMonitor
Rate (kbps)	40	40	40
Frame Rate (fps)	10	10	10
Total Time (ms)	33615	59256	39010

ME /MC (%)	89.30	88.69	82.27
DCT (%)	2.91	4.05	5.37

Table 5.3: MPEG-4 Complexity Results

Again, the computational dominance of the ME/MC routines is clearly evident.

5.2.4 Discussion

The results certainly indicate the computational overhead of motion estimation and compensation. In each case the ME/MC consumed the majority of resources, and in H.263 and MPEG-4, the most recent algorithms, ME/MC accounts for approximately 80% of the computation time.

Thus the greatest savings may be made by improving the ME/MC stage. Even greater computational savings are theoretically possible by not performing ME/MC at all. This was the approach taken, and the task of the following work was to find a suitable way to encode a video sequence without this very popular method.

5.3 The Nature of Difference Frames

Having taken the decision, based on both computational and memory constraints, to encode difference frames (also referred to as difference residual frames), a suitable technique is required. Fortunately, difference frames display strong characteristics in the spatial domain that invite exploitation. This section explores difference frames and the method chosen to encode them.

5.3.1 Intra-Object Correlation

An image is composed of the 2D projection of 3D objects onto the imaging plane. Within the spatial area representing a single object, the interpixel correlation is typically large. This is because objects are typically self-similar, with smooth colour characteristics. The following frames from the test sequences demonstrate this characteristic.

Akiyo #42 News #109 Hallmonitor #114

Table 5.4: Images Showing Local Correlation

Within a physical object, such as Akiyo's jacket, the pixel values are highly correlated. However, the correlation between objects, such as her jacket and hair, is low.

Let the area of pixel correlation, due to the projection of a single object onto the image plane be denoted by:

$$\chi\left(x_i, y_j\right) = \left\{p_{x,y} \mid x, y \in object_i\right\}. \tag{5.1}$$

Here $object_i$ is the 2D projection of a single real world object, $p_{x,y}$ are those pixels at coordinates (x, y) that form the image of the object, $\left(x_{i,}y_j\right)$ is the mathematical centre of the object image. As the region of support (or domain) of the object image is arbitrary, $\left(x_{i,}y_i\right)$ may be defined in any way necessary. Thus $\chi\left(x_i, y_j\right)$ is the set of pixels representing a single object in the image, centred at $\left(x_{i,}y_j\right)$. In this discussion the pixel positions, not values, are under consideration. The value of a pixel is denoted

$$Y(p_{x,y}). \tag{5.2}$$

5.3.2 Object Motion

The second observation is that the correlation within an object's image is not affected by motion. The shirt worn by the man in Hallmonitor does not change colour dramatically as a result of his walking down the passage. This is because the shirt itself is not fundamentally altered by the motion. Of course effects such as lighting and noise detract from this correlation, but are secondary effects. Thus it is fair to say that the area of correlation formed by a body, moves with the body as it translates over the image, due to motion.

A translated image in the plane (due to object motion) may be represented by

$$\chi\left(x_{i+\delta_x}, y_{j+\delta_y}\right). \tag{5.3}$$

Note that equation (5.3) represents the same object as equation (5.1), but in different spatial location. These two sets will only be equal in dimension, in the case of pure image translation. However, it is approximately true for other forms of motion such as rotation or tilting, by noting that the area of support of the function has changed, but the pixel values have not changed dramatically.

Consider two subsequent frames (#n and #(n+1)) in a video sequence. For the case of no global motion, local motion, and no lighting changes or noise, all the changes in pixel values are on account of the local motion of objects. Considering a single object, the pixels that may change on account of this object's motion form a set of the object's initial and final areas of support:

$$\Delta = \left\{\chi\left(x_i, y_j\right) \cup \chi\left(x_{i+\delta x}, y_{j+\delta y}\right)\right\}. \tag{5.4}$$

This set may be alternately written as the union of two other sets:

$$\Delta = \left\{A \cup B\right\}, \tag{5.5}$$

where

$$A = \left\{ \chi\left(x_i, y_j\right) \cap \chi\left(x_{i+\delta x}, y_{j+\delta y}\right) \right\} \tag{5.6}$$

which is the set of pixels where the original and translated images overlap in space,

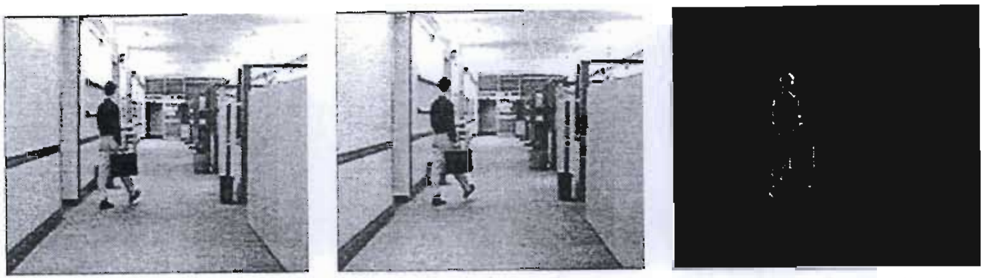
and

$$B = \left\{ \left[\chi\left(x_i + y_i\right) \cup \chi\left(x_{i+\delta x} + y_{j+\delta y}\right) \right] - \left[\chi\left(x_i + y_j\right) \cap \chi\left(x_{i+\delta x} + y_{j+\delta y}\right) \right] \right\}, \tag{5.7}$$

which is the set of pixels where the two images do not overlap in space.

5.3.3 Effect of Object Motion in Difference Frames

The following table shows an example of these sets for the Hallmonitor sequence, considering the moving man as the object in question, correctly the man is a complex object consisting of a head, shirt, pants, briefcase etc.



frame #40 $Y\left(\chi\left(x_i,y_j\right)\right)$

frame #41 $Y\left(\chi\left(x_{i+\delta x},y_{j+\delta y}\right)\right)$

frame #41 – frame #40 $Y(\Delta)$

Table 5.5: Example of Object Motion

The properties of the $Y(\Delta)$ set are distinct for difference frames. Considering the set $Y_{diff}(\mathrm{A})$ is defined:

$$Y_{diff}(\mathrm{A}) = Y_{\#40}\left(p_{x,y} \in A\right) - Y_{\#41}\left(p_{x,y} \in A\right), \qquad (5.8)$$

where the subscript of Y indicates the frame from which the pixel values are drawn. As expressed earlier, the pixels within an object are highly correlated, thus the pixels representing the same object, but in different frames are also highly correlated. The set A consists of these pixels, thus the set $Y_{diff}(\mathrm{A})$ generates small numbers. This is exhibited in Table 5.5 where the interior of the object in the difference frame is dark.

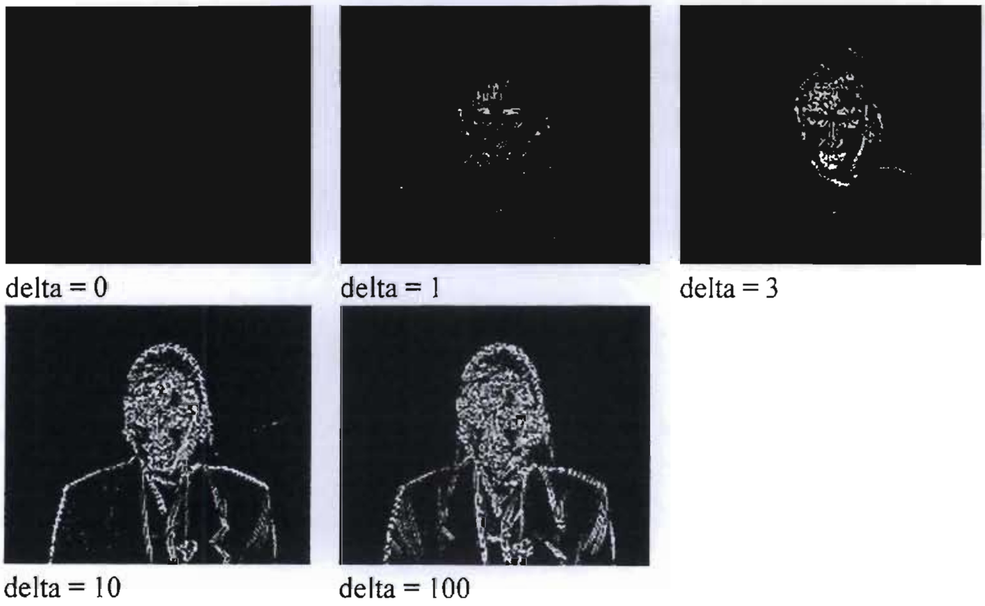
The set $Y_{diff}(B)$ is defined similarly to $Y_{diff}(\mathrm{A})$ by replacing A with B in Equation (5.3). This set behaves differently, as it represents the difference between different objects (usually a moving object and the background). There is no systematic correlation between these objects, thus the difference pixels have high values. Table 5.5 shows this clearly, where all the high value coefficients are on the leading and trailing edge of the moving object, due to the object either covering or uncovering the background.

Thus difference frames remove the stationary background and interior of moving objects from the frame. It shall be shown below that this results in a decrease in the variance of the frame and hence the number of bits required to encode it to within a fidelity criterion.

5.3.4 Spatial Clustering of Significant Coefficients

Another observation is that, owing to this spatial localisation, significant coefficients are spatially clustered in difference frames.

The following image series depicts difference frames around frame 100 of the Akiyo sequence. In each case the image represents $\left|Y_{\#100}-Y_{\#(100+\text{delta})}\right|\cdot 10$. The multiplication by 10 is purely to brighten the image for demonstration.



delta = 0

delta = 1

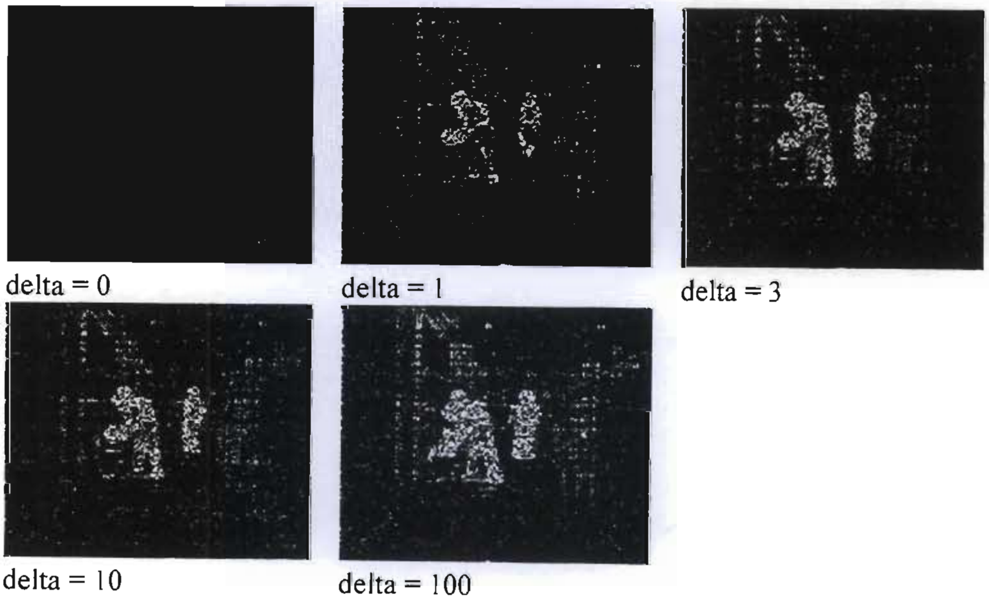
delta = 3

delta = 10

delta = 100

Table 5.6: Difference frames around Akiyo #100

A similar set is shown in Table 5.7 for Hallmonitor, around frame 100.



delta = 0

delta = 1

delta = 3

delta = 10

delta = 100

Table 5.7: Difference Frames around Hallmonitor #100

The effect of interest is the clustering of significant coefficients. As discussed above, the set of pixels in a difference frame that may change as a result of an object's motion is the set

$$\Delta = \left\{\chi\left(x_i, y_j\right)\cup\chi\left(x_{i+\delta x}, y_{j+\delta y}\right)\right\}. \tag{5.9}$$

which is thus localised in space. As a consequence, the set of pixels which are expected to be significant:

$$B = \left\{\left[\chi(x_i + y_j) \cup \chi(x_{i+\delta x} + y_{j+\delta y})\right] - \left[\chi(x_i + y_j) \cap \chi(x_{i+\delta x} + y_{j+\delta y})\right]\right\}, \qquad (5.10)$$

is also spatially localised.

This effect is referred to as coefficient clustering and has been shown to be a fundamental property of difference frames for local motion video sequences. This clustering is in effect a statistical stationarity, and experience from previous image and video compression schemes indicates that such stationarity may be exploited to obtain coding advantage.

5.3.5 Rate Distortion Behaviour of Difference Frames

In Chapter 2, Shannon's rate distortion equation was shown to be

$$R(D) = \max\left[0, \frac{1}{2}\log\left(\frac{\sigma^2}{D}\right)\right] \qquad (5.11)$$

for Gaussian sources. Thus to minimise the rate, subject to a distortion constraint, the variance must be minimised. The following sections shall show that difference frames have a lower variance than the original frames (Section 5.3.5.1), and how coefficient clustering may be exploited to gain further compression advantages (Section 5.3.5.2).

5.3.5.1 Variance of Difference Frames

The following table shows the effect of difference frames on the source variance:

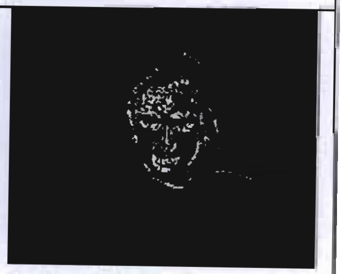
Source Label	Source 1	Source 2
Source	 Frame #103	 Frame #103 – Frame #100
Mean	92.69	0.766
Variance (σ^2)	2234.53	7.43
$\frac{1}{2}\log_2(\sigma^2)$	5.56	1.45

Table 5.8: Variance of Original and Difference Frames

Assume initially the image to be modelled is a Gaussian source, which is commonplace, and justified in [Dono98]. Indeed, that the DCT approximates the KLT (Karhunen Loeve Transform), for Gauss-Markov processes is the often cited reason why JPEG and MPEG techniques rely on the DCT [Dono98]. This assumption is made for the purposes only of the following development, and is abandoned in favour of more accurate methods in the implemented algorithm (Chapter 6).

According to the Shannon bound, (5.11), Source 1 requires more bits than Source 2, to achieve an equal distortion, D, specifically:

$$\begin{aligned} R_1(D) - R_2(D) &= \frac{1}{2}\left(\log_2\left(\frac{\sigma_1^2}{D}\right) - \log_2\left(\frac{\sigma_2^2}{D}\right)\right) \\ &= \frac{1}{2}\left(\log_2\left(\frac{\sigma_1^2}{\sigma_2^2}\right)\right) \\ &= 5.58bpp \end{aligned} \qquad (5.12)$$

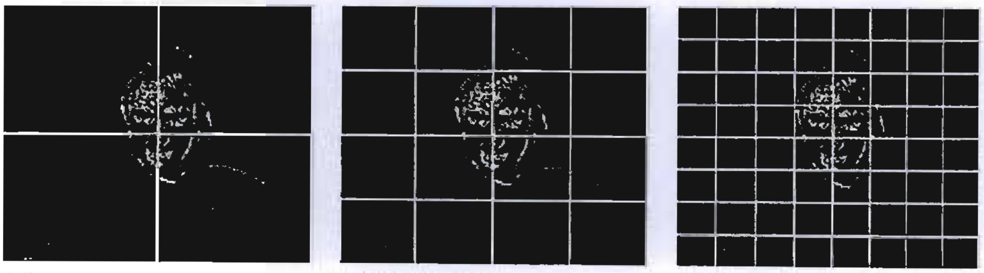
This equation only hold for cases where $\frac{\sigma^2}{D} \geq 1$.

Although somewhat contrived, this example shows how difference frames increase the theoretical compression ratio over the original image, as a result of their small variance.

5.3.5.2 Source Partitioning to Exploit Coefficient Clustering

The previous section has shown that reducing the variance of a source reduces the number of bits required to code it, and that difference frames function to achieve this. In addition to this, the clustering behaviour of difference frames may be used to achieve further gains. This spatially varying behaviour suggests a spatial source set partitioning an in [Lin01].

Consider the partitioning of Source 2 into a various number of tiles:



4 Partition 16 Partition 64 Partition

Table 5.9: Source Partitioning

Intuitively, it seems that coding each independently may yield advantage. The following development investigates this possibility.

5.3.5.2.1 Derivation

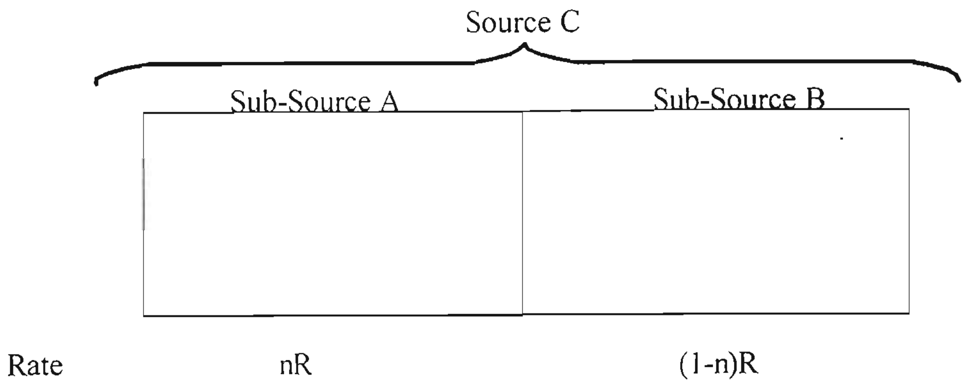


Figure 5-1: Source Partitioning Example

Figure 5-1 depicts a generic image source, Source C, containing N pixels. Source C has been partitioned equally into two subsources, A and B, such that both N/2 pixels. It is assumed that A, B and C satisfy the requirements of having a Gaussian distribution, and flat spectral density, this enables Shannon's analytical machinery. The question to be addressed below is whether coding A and B independently will yield a lower distortion to coding C, if the same number of bits is used in each case.

If Source A and B are coded with a total rate, R bpp, divided between them such that A receives nR and B receives (1-n)R bpp, then the total bits required is

$$nR\cdot\frac{N}{2}+(1-n)R\cdot\frac{N}{2}=\frac{R\cdot N}{2}\text{ bits}\qquad n\in\left[0,1\right].\tag{5.13}$$

In order for Source C to be coded with the same number of bits, a rate of:

$$\frac{R\cdot N}{2}\cdot\frac{1}{N}=\frac{R}{2}\text{ bpp}\tag{5.14}$$

must be used.

In order for the source partitioning to yield an advantage, the total distortion of the partitioned scheme must be less than that of the original scheme, that is

$$\begin{gathered}D_C\cdot N-D_A\cdot\frac{N}{2}-D_B\cdot\frac{N}{2}<0\\ D_C-\frac{1}{2}\left(D_A+D_B\right)<0\end{gathered},\tag{5.15}$$

where by convention D_C represents the average distortion of.source C.

Maintaining the Gaussian assumption allows (5.15) to be written in terms of the rate:

$$\Delta D=\sigma_C{}^2\cdot 2^{-2(R/2)}-\frac{1}{2}\left[\sigma_A{}^2\cdot 2^{-2nR}+\sigma_B{}^2\cdot 2^{-2(1-n)R}\right].\tag{5.16}$$

The maximum value of this function will occur when its derivative with respect to n, the only free variable at this point, is zero:

$$\frac{d\Delta D}{dn} = 0 - \frac{1}{2}\sigma_A^{\,2} \cdot 2^{-2nR} \cdot \ln 2 \cdot (-2R) - \sigma_B^{\,2} \cdot 2^{-2(1-n)R} \cdot \ln 2 \cdot 2R \;= 0$$

$$\sigma_A^{\,2} \cdot 2^{-2nR} = \sigma_B^{\,2} \cdot 2^{-2R} \cdot 2^{2nR} \tag{5.17}$$

$$\therefore \quad \log_2\left(\sigma_A^{\,2}\right) - 2nR \cdot \log_2 2 = \log_2\left(\sigma_B^{\,2}\right) - 2R \cdot \log_2 2 + 2nR \cdot \log_2 2$$

$$\log_2\left(\frac{\sigma_A^{\,2}}{\sigma_B^{\,2}}\right) = 4nR - 2R$$

The value of *n* yielding this optimal arrangement is thus:

$$\tilde{n} = \frac{1}{2} + \frac{\log_2\left(\dfrac{\sigma_A^{\,2}}{\sigma_B^{\,2}}\right)}{4R} \tag{5.18}$$

The following ratios are defined:

$$k = \frac{\sigma_A^{\,2}}{\sigma_B^{\,2}}, \qquad l = \frac{\sigma_A^{\,2}}{\sigma_C^{\,2}} \tag{5.19}$$

Substituting (5.19) into (5.16)and simplifying:

$$\begin{aligned}\Delta D &= \sigma_C^{\,2} \cdot 2^{-R} - 2^{-1}\left(\sigma_A^{\,2} \cdot 2^{-2nR} + \frac{\sigma_A^{\,2}}{k} \cdot 2^{-2R+2nR}\right) \\ &= \sigma_C^{\,2} \cdot 2^{-R} - 2^{-1}\sigma_A^{\,2} \cdot 2^{-2nR}\left(1 + \frac{1}{k}2^{-2R+4nR}\right) \\ &= \sigma_C^{\,2} \cdot 2^{-R}\left[1 - l \cdot 2^{R-2nR-1}\left(1 + \frac{1}{k}2^{-2R+4nR}\right)\right] \\ &= \sigma_C^{\,2} \cdot 2^{-R}\left[1 - l \cdot 2^{R-2nR-1} + \frac{l}{k} \cdot 2^{-R+2nR-1}\right]\end{aligned} \tag{5.20}$$

Equation (5.20) gives a decrease in average distortion, for any value of n. The greatest achievable decrease in average distortion is found by substituting (5.18) into (5.20):

$$\Delta D = \sigma_C^{\,2} \cdot 2^{-R}\left[1 - l \cdot 2^{R-2\tilde{n}R-1} - \frac{l}{k} \cdot 2^{-R+2\tilde{n}R-1}\right]. \tag{5.21}$$

To simplify the working the terms involving $\tilde{n}$ are considered separately:

$$\begin{aligned} 2^{-2\bar{n}R} &= 2^{-2\left(\frac{1}{2}+\frac{\log_2(k)}{4R}\right)R} \\ &= 2^{-R}\cdot 2^{-\frac{1}{2}\log_2 k} \\ &= \frac{1}{2^R\cdot\sqrt{k}} \end{aligned} \quad . \tag{5.22}$$

Equivalently:

$$2^{+2\bar{n}R} = 2^R\cdot\sqrt{k}\ . \tag{5.23}$$

Substituting (5.22) and (5.23) back into (5.21) yields:

$$\begin{aligned} \Delta D_{\max} &= \sigma_C{}^2\cdot 2^{-R}\left[1 - l\cdot 2^{R-1}\cdot\frac{1}{2^R\sqrt{k}} - \frac{l}{k}\cdot 2^{-R-1}\cdot 2^R\sqrt{k}\right] \\ &= \sigma_C{}^2\cdot 2^{-R}\left[1-\frac{l}{2\sqrt{k}}-\frac{l}{2\sqrt{k}}\right] \\ &= \sigma_C{}^2\cdot 2^{-R} - \sigma_C{}^2\cdot\frac{\sigma_A{}^2}{\sigma_C{}^2}\cdot\frac{\sigma_B}{\sigma_A} \\ &= 2^{-R}\left(\sigma_C{}^2-\sigma_A\sigma_B\right) \end{aligned} \quad . \tag{5.24}$$

This is an extremely pleasant result, which indicates that at a particular rate R, a source partitioning with optimal bit allocation will yield a distortion decrease which is a function of the variances.

If source C can be partitioned such that the variance of either or both of A and B is significantly less than the variance of C, an advantage may be gained. This behaviour may be expected of a source with strong local stationarities. As mentioned in Chapter 3 (Section 3.3) images display exactly this property.

Comparing equation (5.24) with (5.16) it is apparent that an even source partition with optimal bit allocation will yield a distortion of:

$$2^{-2\frac{R}{2}}\sigma_A\sigma_B\,, \tag{5.25}$$

where $\frac{R}{2}$ is the total bit rate.

This analysis shows that the critical factor allowing advantage from source partitioning is a spatial concentration of variance. This is a statistical stationarity, and clearly it has been exploited.

5.3.5.2.2 *Local Stationarity*

Consider two adjacent tiles from the 16 way partitioned image in Table 5.9.

Source	Source C	Source A	Source B
Image			
Mean	22.46	38.38	11.18
Variance	2215.82	2609.58	1243.32

Table 5.10: Example Partitioning

Adopting the symbols from above:

$$\begin{aligned} \sigma_C^2 &= 2215.82 \\ \sigma_A \sigma_B &= 1801.26 \end{aligned}$$

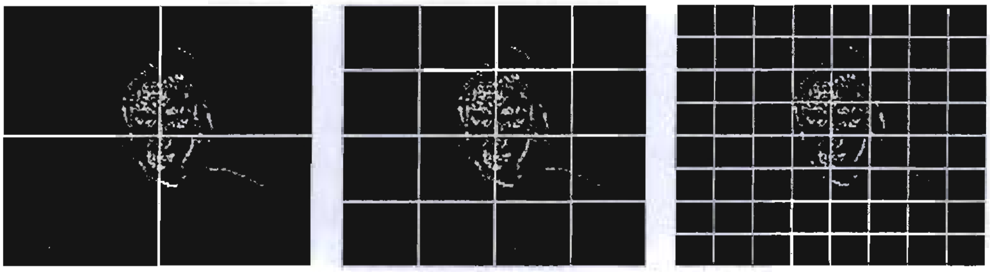
Hence, equation (5.24) predicts that partitioning will yield a gain of

$$\begin{aligned} \Delta D_{\max} &= 2^{-R}\left(\sigma_C^2 - \sigma_A \sigma_B\right) \\ &= 2^{-R}(414.56) \end{aligned} \qquad (5.26)$$

This is not entirely accurate, as the actual images shown above do not necessarily conform to Shannon's Gaussian assumptions. However, these measurements do provide the expectation of a coding gain, should proper rate-distortion estimation be applied to the tiles of the partitioned image.

5.3.5.2.3 Multiple Partitions

Although the maximum gain shown above may seem marginal, partitioning the source multiple times compounds the effect, provided that each partition acts to concentrates the variance. Presenting Table 5.9 once again:



4 Partition 16 Partition 64 Partition

Table 5.11: Source Partitioning

In the case shown above the source is partitioned by a factor of two in each dimension at each iteration. At each partition stage, a possible performance gain may be made, if partition serves to concentrate the source variance. These gains are additive, thus as the number of partitions increase, so does the gain, for an amenable source.

probability of a significant coefficient is dependant on the significance of its spatial, temporal and spectral neighbours.

Section 5.3.5.2 develops an argument based on Shannon's theory, indicating how source partitioning and optimal bit allocation may be employed to exploit the local stationarity to increase the coding performance of difference frames.

The conclusion reached is that difference frames may be successfully coded using a suitable partitioning and bit allocation method. The EBCOT and JPEG-2000 (Section 3.3.9) still image compression algorithms operate on a tile partitioned source, with optimal bit allocation. Still images exhibit significantly less clustering than difference frames and source partitioning has clearly been found to be of utility. Hence we expect this partitioning gain to be greater for difference frames.

The performance advantage of such a system over traditional ME/MC and 3D coding schemes should be handsome. Indeed such a computational advantage, rather than compression performance, is a project imperative. For this reason a video compression system based on the above philosophy is developed in the following Chapter.

Chapter 6 - Proposed Algorithm

This chapter proposes a new video compression algorithm. The major requirement of this algorithm is low complexity, to allow later embedded implementation. Chapter 5 has shown that significant complexity saving can be made through abandoning block ME/MC, but impact of this on the RD performance is expected to be dire. Chapter 5 has proposed difference frames as an alternative to block ME/MC, and presented a model of these frames. This chapter also showed that source partitioning will allow efficient compression of these frames. The goal of this chapter is to design a new video compression scheme, based on this source model, which presents low complexity as well as acceptable RD performance.

Chapter 5 furthers the argument for difference frames by illustrating that source partitioning offers the opportunity to increase the RD performance when coding such frames. Unfortunately the argument presented there assumes that the difference frames exhibit a Gaussian distribution, which is clearly an oversimplification for real sources. Thus the source estimation and bit allocation routines presented in Chapter 5 must be replaced with accurate and robust methods that do not rely on this Gaussian assumption. This problem is addressed in this chapter, thus realising the temporal decorrelation scheme postulated in Chapter 5.

Thereafter, both a spatial coding and an entropy coding stage must be designed. This design process is also described below. Thus, this chapter describes the design of the complete compression algorithm, based on the decision to use difference frames and source partitioning.

Section 6.1 will detail the algorithm, around a block-diagram design. For each stage in the system, the choice of algorithm will be justified, based on the preceding literature review. This will provide a broad view of the implemented system.

Thereafter Section 6.2 will describe the software level design of the system. This will investigate each stage in detail, presenting and justifying any departures from the literature. Interim results will be presented throughout this discussion, indicating how various design decisions were made.

6.1 Algorithm System Design

6.1.1 Overview and General Architecture

This section will present a general block diagram of an appropriate video compression architecture. The requirements and algorithm choices for each block will then be made and justified.

The various stages of the encoder side of the system may be represented by the following block diagram:

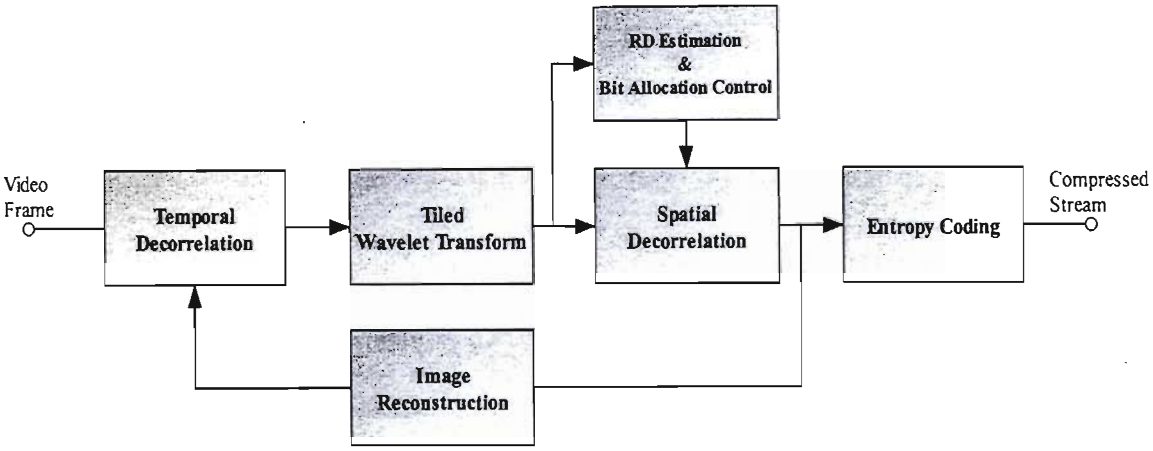


Figure 6-1: Video Encoder Block Diagram

Initially, temporal decorrelation is performed. This is achieved by producing a difference frame by subtracting a reconstruction of the previously coded frame from the incoming frame. This difference frame is input to the wavelet transform stage. The wavelet transform stage segments the frame into tiles and applies the 2D wavelet transform to the each tile independently. The entire tiled and wavelet transformed difference residual frame is then input to the RD Estimation block. This block examines the residual frame and estimates the RD behaviour of each tile. Based on this, an optimal bit allocation between the tiles is calculated. This optimal bit allocation is used to drive the Spatial Decorrelation block. This block codes each image tile, to produce an output stream of the specified bit length. The output stream of each tile is concatenated and input to the Entropy Coder. This performs adaptive arithmetic coding to reduce the bit rate in a lossless fashion. This stream is then output.

The decoder is somewhat simpler.

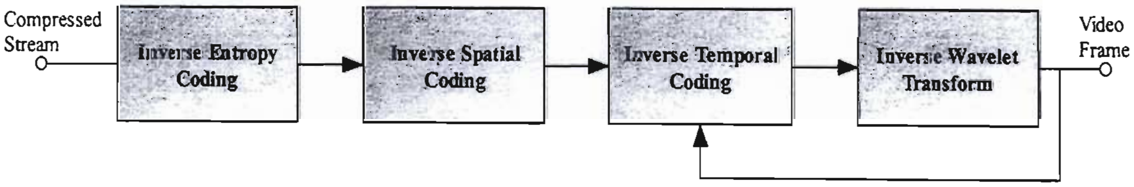


Figure 6-2: Video Decoder Block Diagram

The encoded frame is sequentially arithmetically decoded, inverse spatially compressed, added to the previous output frame, and inverse wavelet transformed, to produce a representation of the original frame.

The following discussion will examine each stage of this process in more detail, and justify the choice of implementation algorithm chosen in each case.

6.1.2 Temporal Decorrelation Stage

Temporal Decorrelation is achieved through frame differencing. As fully expressed in Chapter 5, this method was chosen owing to its very large computational advantage over any other method available. This choice is the fundamental departure of our algorithm from the norm and sets the challenge for the remaining stages to present acceptable RD performance.

6.1.3 Transform Stage

By definition, the project is based on the wavelet transform, thus limiting the design choices for this stage of the system. However, as Section 3.2 indicates, there are several degrees of freedom in the choice of both the mother wavelet function, and decomposition tree.

Several mother functions have been proposed in the literature for image coding. However, as [Saha99] shows (Section 3.2.1.1) the performance of any of these functions over the standard Daubechies 9/7 wavelet is marginal. This suggests that either departing from this standard basis function, or introducing adaptivity, will not yield substantial gain.

The wavelet packet decomposition explored in Section 3.2.1.3 presents good performance. However, results are only given for the 'Barbara' test image, and thus this good performance in general cannot be guaranteed, especially as the 'Barbara' image is characterised by unusually high high-frequency content. The algorithm itself is described as 'fast' but still represents a complexity increase over a standard dyadic tree, without a guaranteed performance increase. Therefore, this algorithm was not implemented.

Other more recent innovations, such as multi-wavelets and non-separable 2D bases (Sections 3.2.2 and 3.2.3 respectively) have not yet delivered a performance gain for natural images. Thus they are not suitable for our application.

For the reasons given above the non-adaptive, Daubechies 9/7 wavelet was chosen as the basis function, with the standard dyadic decomposition tree.

The number of decomposition levels used is left as an implementation variable. The number of levels possible depends both on the image size. Thus, as an image segmentation algorithm (Section 6.1.4) is employed, this variable has to be available.

6.1.4 RD Estimation Stage

Section 5.3.5.2 presents analytical evidence that source partitioning with optimal bit allocation provides strong coding advantage for difference frames. The analytical framework relies on the assumption that the source adheres to the Gaussian distribution. However, this assumption may prove false in practice and another means of source modelling and subsequent optimal bit allocation must be employed.

Thus the extremely successful method of He and Mitra, [He01] (described fully in Section 2.1.4) is adopted for source modelling. This method is both computationally modest and highly accurate, which motivates for its use. This RD estimation algorithm is applied independently to each tile to obtain its R(D) curve.

Based on this estimation set, a numerical bit allocation algorithm, described Section 6.1.5, is implemented. This algorithm is based on the theoretical development of Section 2.2.

With these two methods the gains of source partitioning may be realised, without the hazard of the Gaussian assumption proving false.

He and Mitra apply the ρ-domain method to entire frames to obtain their RD behaviour. It was unknown whether this method could be applied to much smaller image tiles and still obtain good performance. Furthermore He and Mitra do not consider either wavelet video coding, or difference frame coding, thus this work represents the first application of this method to such coding methods. Fortunately, as this method is based on multiple linear regressions, it is successfully adapted to this new data set.

6.1.5 Optimal Bit Allocation Stage

Having obtained the set of R(D) curves for each tile, an optimal bit allocation must be calculated. Here the method of Lagrange multipliers is employed, as described in Section 2.2. This method is both fast and, as shown previously, converges to optimality. Furthermore, Chapter 3 indicates that many of the published algorithms utilise Lagrange multipliers for a range of optimisation problems, which further motivates for its use.

The Lagrange method will ensure that each tile operates at equal slope on the R(D) curve. This has been shown to be the optimal state, for a given rate. However, matching this required slope, or value of the Lagrange multiplier, to the rate is another problem. In [Ram93], Ramchandran demonstrates a numerical bisection method to solve a similar problem during tree pruning for optimal wavelet packet design (Section 3.2.1.3). This method has been adopted and modified to suit our problem.

Thus, this stage matches the desired rate for a particular frame with the required R(D) curve slope, and this to the bit rate for each tile. Thus the output of this stage is the optimal bitrate for each tile in the frame.

6.1.6 Spatial Coding Stage

This stage applies a still image compression technique to each tile independently. Chapter 3 provides a review of all the methods that were considered. A key requirement of the algorithm is the production of an embedded bitstream. The Optimal Bit Allocation stage specifies the optimal number of bits to code each tile, thus the ability to code to any desired bitrate is important.

The first observation from the results presented in Section 3.4 is that the coding performance difference between any of the algorithms is, at best, marginal. At the low bit rates of interest, most algorithms perform to within 1bB of another. The computational complexity of each algorithm varies greatly, however it is hard to ascertain precisely as numerical results are mostly unavailable. Thus, informed guesswork is required to estimate the complexity. The following brief argument discusses the suitability of each algorithm in the literature review.

EZW (Section 3.3.3) was discounted as it is the worst performing algorithm, without significant computational advantages.

SPIHT (Section 3.3.4) combines reasonable performance with computational modesty, thus is a contender. The RD optimisation of SPIHT (Section 3.3.5) presents no performance gain at low bitrates, thus was disregarded.

SFQ (Section 3.3.6) is one of the most computationally severe algorithms presented due to the multiple iterations required for the tree search, and joint optimisation of zerotree quantisation criterion and quantiser stepsize. As complexity is crucial this method was not considered.

SRC (Section 3.3.7) suffers both higher implementation cost, and poorer performance than SPIHT, thus was abandoned.

TCQ (Section 3.3.8) presents inferior performance to the benchmark SPIHT algorithm, as well as undefined computational demand. Thus it could not be considered.

ECECOW (Section 3.3.9) presents good performance and low computational demands, thus was considered.

EQ (Section 3.3.10) does not have published computational complexity, although it is expected to be high. Thus although the algorithm exhibits good performance, the unknown complexity excludes it from consideration.

TCSFQ (Section 3.3.11) is the most computationally demanding algorithm evaluated, combining trellises, SFQ, and conditional entropy coding. This was far beyond the allowed computational load, and was deemed unsuitable.

EBCOT (Section 3.3.12) is designed for general image processing, and contains many features that are irrelevant to our application. The rate cost imposed by these features detracts significantly from its performance. Thus the algorithm is both complex and underperforming, and was not considered.

Thus only two algorithms satisfy both the computational and performance requirements; SPIHT and ECECOW, both of which satisfy the requirements of generating an embedded bit stream. The issue which remains is a consideration of each algorithm's expected ability to represent difference residual frames. SPIHT has been applied, with modification, to the coding of ME/MC residual frames (Section 4.3.3.3), as well as to 3D coding in 3D-SPIHT (Section 4.3.5.1). The 3D-ESCOT algorithm (Section 4.3.5.2), is based on EBCOT, which has a similar operating principle to ECECOW. This algorithm performs very well, however, this author believes the performance advantage is imparted by the tile based, RD optimal bit allocation that 3D-ESCOT employs, rather than the context adaptive arithmetic coding, which is the common principle with ECECOW. As our algorithm already employs a tile based bit allocation strategy this possible gain is already accounted for. These factors motivate for the use of SPIHT.

The published ECECOW algorithm utilises a context formation strategy that is based on an empirical study of still images. Thus although the concept of context formation for adaptive arithmetic coding is still valid for residual images, their differing statistics necessitates redesigning the context formation stage. The zerotree premise of SPIHT relies on energy concentration in the lower subbands, as well as clustering of insignificant coefficients, spatially to facilitate later arithmetic coding, and across bands to facilitate zerotree quantization. It is believed that conditions will hold for difference residual frames, indeed the spatial clustering has already been demonstrated in Chapter 5.

Finally the author is already in possession of an understanding of SPIHT as well as source code for the algorithm as a result of a commercialisation exercise involving SPIHT for still image coding, based on the work of [McIn02]. Thus, modifying this code for use in this project represents a significant time saving, allowing further effort to be expended in other areas, such as the RD estimation and optimisation, which are the more novel aspects of this work.

For all of these reasons it was decided to implement the spatial coding of the tiles using the SPIHT algorithm. It was noted, however, that future work could be based on an ECECOW algorithm, with a redesigned context formation module, as would enable better source modelling to be achieved.

6.1.7 Entropy Coding Stage

The final stage is the entropy coder. The choice for this stage falls essentially between either a Huffman or an arithmetic coder. It was decided again to revert to the previous work of [McIn02]. In this work an adaptive arithmetic coder was successfully applied, for which the source code was available. As [McIn02] finds the scheme effective for SPIHT still frame coding, and our system is also based on SPIHT, it is expected to remain effective. Furthermore as the focus of the project is on the temporal decorrelation stage, it was time effective to employ this coder.

6.1.8 Summary of System Design

The previous sections have given an overview of the system design process, and resulting algorithm. The core decisions were to use the wavelet transform and difference frames for temporal decorrelation. Difference frames were chosen for their computational advantages. Having taken this decision, a means of achieving competitive compression performance had to be found. Based on the study in Chapter 5, it was believed that source partitioning with optimal bit allocation would provide such a means. In order to assure optimal bit allocation, the ρ-domain RD method of He and Mitra was incorporated as the foundational technique of the system. The RD estimate generated by this stage drives the optimal bit allocation control algorithm, which in turn controls the spatial coder. Clearly the accuracy of the RD method is crucial to the success of this algorithm.

Through tiling based source partitioning and RD estimation, the local spatial stationarities of difference frames, as explored in Chapter 5, are exploited. This is the means of temporal decorrelation. Spatial decorrelation is achieved through individual tile coding using SPIHT. This exploits the joint spatial and spectral clustering of insignificant coefficients, as well as the coefficient energy concentration in the low frequency band. This is the spatial decorrelation method. Finally the concatenated output stream of all the tile coders is arithmetically coded to reduce the source entropy.

Thus the use of these various stages accounts for temporal correlation between frames, the spatial behaviour of images and source entropy reduction. These are arguably the main factors enabling any compression system.

Thus our algorithm my be classified as RD optimised adaptive coder. The prolific researcher Z. Xiong notes in [Xiong99,2], "The most competitive results seem to be generated by either RD optimised classification and quantization ... or efficient context based modelling." That our enquiry led to such a system is promising.

In addition, it is clear that our algorithm is designed to be computationally modest. The wavelet transform is a fast process; temporal decorrelation through difference frames is the fastest such technique; SPIHT is one of the simplest wavelet quantisers in the literature; the ρ-domain RD estimation method is extremely fast, relying only on several multiplications; the optimal bit allocation consists mainly of a bisection search, which is a very fast method and finally the arithmetic coder is also known to be a computationally simple process. Thus, although there are several stages in the algorithm, each is computationally efficient and the algorithm may be realistically expected to have fast execution times. This expectation is confirmed in Chapter 7.

6.1.9 System Block Diagram

The entire system may now be presented in detail, in the form of the following block diagram.

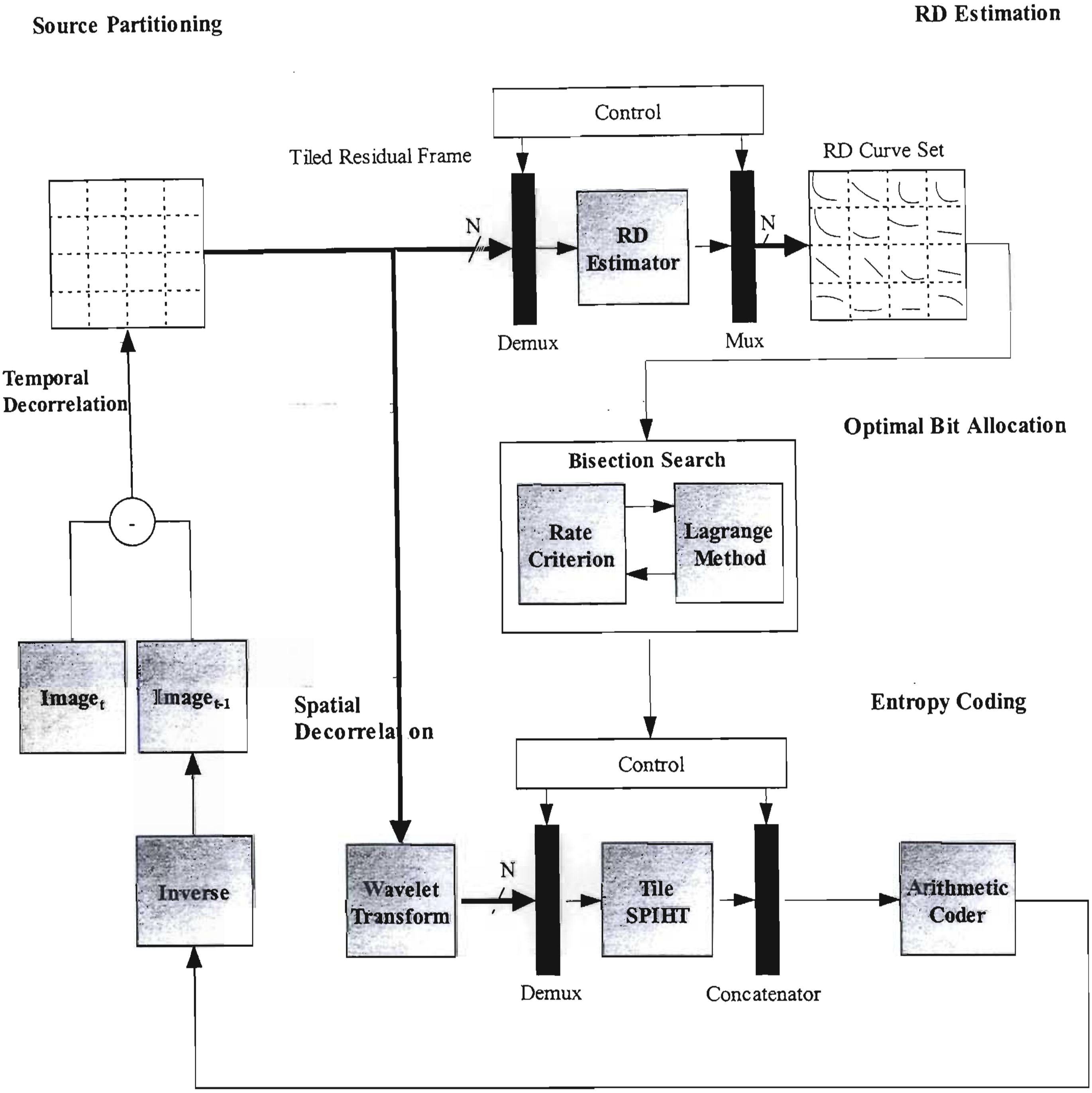


Figure 6-3: System Block Diagram

6.2 Algorithm Implementation

The previous section reports on the philosophy and design of the overall system, as well as the choice of algorithm for each stage. This section will now provide further detail on each stage and indicate implementation level details. Departures from the literature were made and these will be highlighted as necessary.

6.2.1 Temporal Decorrelation Stage

As mentioned, temporal decorrelation is achieved through de-compressing the output stream of the previous frame and subtracting it from the current frame.

Using the decompressed previous frame is important. Although using the original, previous frame saves the computational overhead of decompression, it results in unacceptable coding performance. High-ratio compression is necessarily lossy, and in many cases this loss is greater than the difference between subsequent frames. If subsequent original frames are used for differencing, this error is never accounted for, and thus grows without bound. Using the decompressed previous frame for differencing is a feedback mechanism that serves to minimise the total error of output frames.

6.2.2 Transform Stage

The wavelet transform is achieved using the lifting methodology developed by W. Sweldens. This technique allows both a fast implementation on hardware, as well as minimising the number of floating point operations necessary to maintain a lossless transform.

The code used was developed by I. MacIntosh in [McIn03], and is used without significant changes.

6.2.3 RD Estimation Stage

Adapting and implementing the RD estimation stage of the algorithm required some work. The original, and then only, work using the ρ-domain method was that of He and Mitra [He01] and [He01,2]. In this work they consider the RD estimation firstly of still frames, and then motion compensated frames in DCT based video systems. The latter is considered for the purposes of frame-by-frame rate control. Our data is different; sub-frame datasets from wavelet difference residual frames. This data is different; wavelet transformed difference frames are expected to behave differently to DCT encoded, motion compensated residual frames. Furthermore, the tiles are significantly smaller than the original frame and as the ρ-domain method is built on statistical regression, there was concern that the sample size would be too small to render meaningful statistical description.

The software was developed in two stages. The first was a development and testing of the RD scheme on still images. This was undertaken to test the functionality of the software, as He and Mitra's results for this experiment are available for comparison. The second stage was to adapt the method for the difference frame tiles.

6.2.3.1 Still Image RD Estimation

The following set of images was considered. This set was chosen for its diversity, including natural, and synthetic images of natural, technical, and artistic scenes. Several standard images are included.

Figure 6-4: RD Estimation Regression Set

This set of images was used to perform the multiple regressions in order to configure the ρ-domain method.

6.2.3.1.1 Qz and Qnz Calculation

Following Section 2.1.4, the first element of configuration is the calculation of the pseudo-coding bitrates. A vector of test values of ρ was defined:

$$\rho = [0.7, 0.75, 0.8, 0.85, 0.9, 0.95].$$

For each value of ρ_i, $Q_{nz}(\rho_i)$ was calculated according to 'first principles.' Although He and Mitra define (Chapter 2, equation 2.18):

$$Q_{nz}(\rho) = \frac{1}{N}\sum_{\forall x\neq 0}\left\lfloor \log_2 I_\rho(x)\right\rfloor + 2, \tag{6.1}$$

this does not reflect the progressive bitscan encoder employed by SPIHT. $Q_{nz}(\rho_i)$ is intended to represent the pseudo-bitrate of the non-zero coefficients, and thus calculates the number of bits required to represent each such coefficient. Although accurate for DCT based schemes such as JPEG and H.263, SPIHT is different. The image is scanned in a progressive bitplane order, thus for a given value of ρ, only the bitplanes above the current have been output. Thus the output stream representation of those coefficients significant at a certain threshold is

$$I(x) - \mathit{threshold}.$$

Hence equation (6.1) is modified:

$$Q_{nz}(\rho)=\frac{1}{N}\sum_{\forall x\neq 0}\left\lfloor \log_2 I_\rho(x)-threshold\right\rfloor+2\,. \tag{6.2}$$

$Q_z(\rho)$ is calculated according to the original work:

$$Q_z=\frac{1}{M}\sum_{zeroruns}\left\lfloor \log_2(runlength)\right\rfloor+2 \tag{6.3}$$

In order to generate each of the pseudo bit rates, the following C-code is executed.

```
                                        /* decrement the number of zeroes */

                              if (run_length != 0)
                                   /* log 0 will return infinity, so avoid this case */

                              Qz +=  floor( log(run_length)/log(2) ) + 2;
                              run_length = 0;
                    /* increment Qnz*/
                              if (abs(Image[y][x]) == threshold)
                                   Qnz +=2;
                              else
                                   /*calc Qnz as per equation (6.2) */
Qnz += floor(log(abs(Image[y][x])-threshold)/log(2)) + 2;
                         }
                    }
          }

          /* Change the counters to ratios */
          p_q /= (y_dim* x_dim);
          Qz /= (y_dim * x_dim);
          Qnz /= (y_dim * x_dim);

          /* Decrease the threshold in preparation for another raster scan */
threshold /= 2;
} /* end raster scan*/
} /* end desired_p interation
```

Code Fragment 1: Qz and Qnz from First Principles

The purpose of this code is to find the state of the SPIHT coder at a particular value of ρ. By mimicking the decreasing threshold (bitplane scan order) scan of the SPIHT coder, the significant and insignificant coefficients at a particular threshold are found. From these, the introduced distortion and Qz and Qnz are calculated. Finally kappa is calculated. All these variables are then written to an output file, for offline consideration.

Although the bitplane traversal mimics SPIHT's behaviour, the raster scan is not accurate, as SPIHT descends significant trees to find further significant coefficients. It was decided that this behaviour was too specific and complex for our purposes, and was not included.

The curves for Qz and Qnz using this first principles method is shown in Figure 6-5.

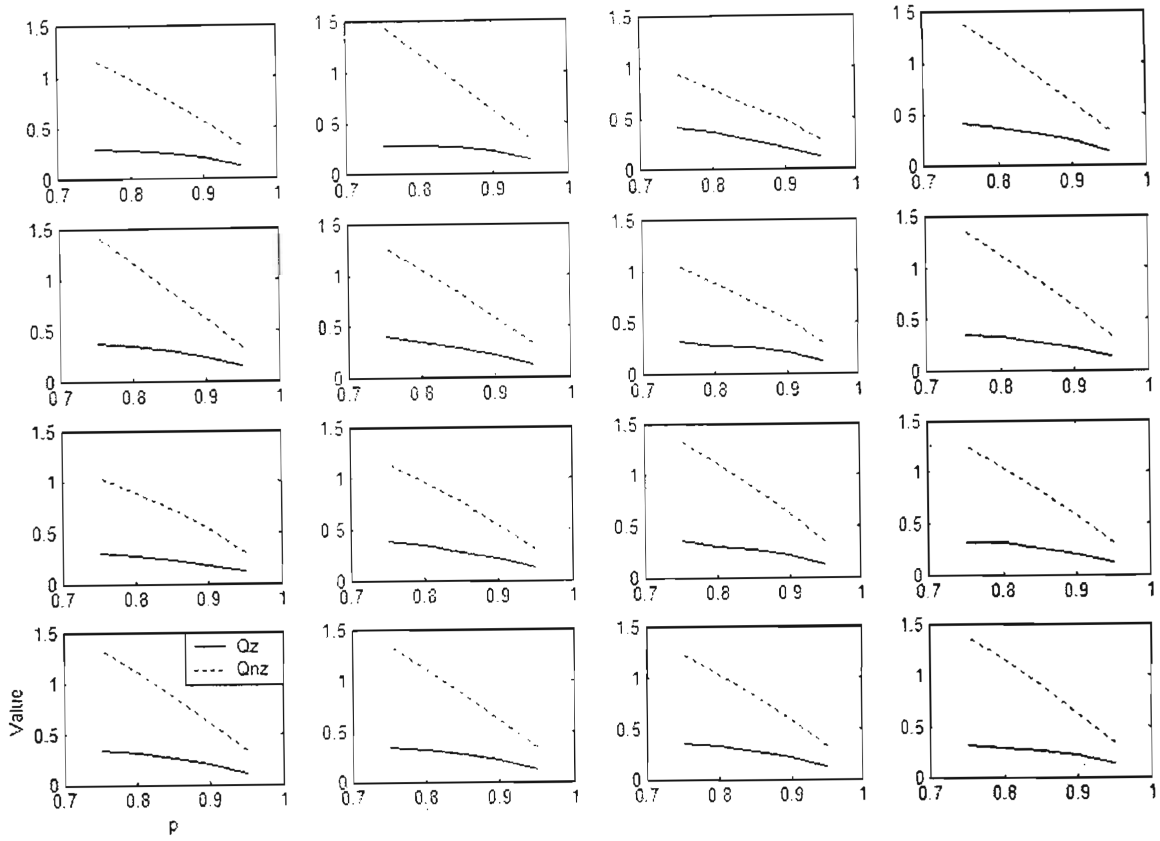


Figure 6-5: Qz and Qnz from First Principles

Here each curve represents the data obtained from the image shown in the respective position in Figure 6-4. These curves clearly indicate the near linearity of Qnz and the flatter response of Qz, as exhibited in He and Mitra's work (Figure 3 in Chapter 2). It is striking how such different images produce such similar curves.

6.2.3.1.2 κ-Qz Regression

Having found the curves from first principles, it remains to configure the fast estimation procedures.

The first parameter required is κ, the gradient of Qnz. This was found simply by calculating

$$\kappa = \frac{Qnz}{1-\rho} \tag{6.4}$$

at $\rho = 0.8$; the middle of the line. This allows Qnz to be estimated as a straight line with this gradient. As this is a fast process, it is performed in the actual algorithm, on a per image basis.

As Qz is more complex, it is estimated using an offline regression, relating the value of Qz at various value of ρ, to κ,

$$Q_z(\rho_i) = A_i\kappa + B_i\ , \tag{6.5}$$

as fully expressed in Section 2.1.4.

The following plots are arranged as in Chapter 2, Figure 6. Each shows a plot of Qz(ρ) vs κ for the 16 test images. Each plot is for a different value of ρ. A least mean squares fitted straight line curve is also shown.

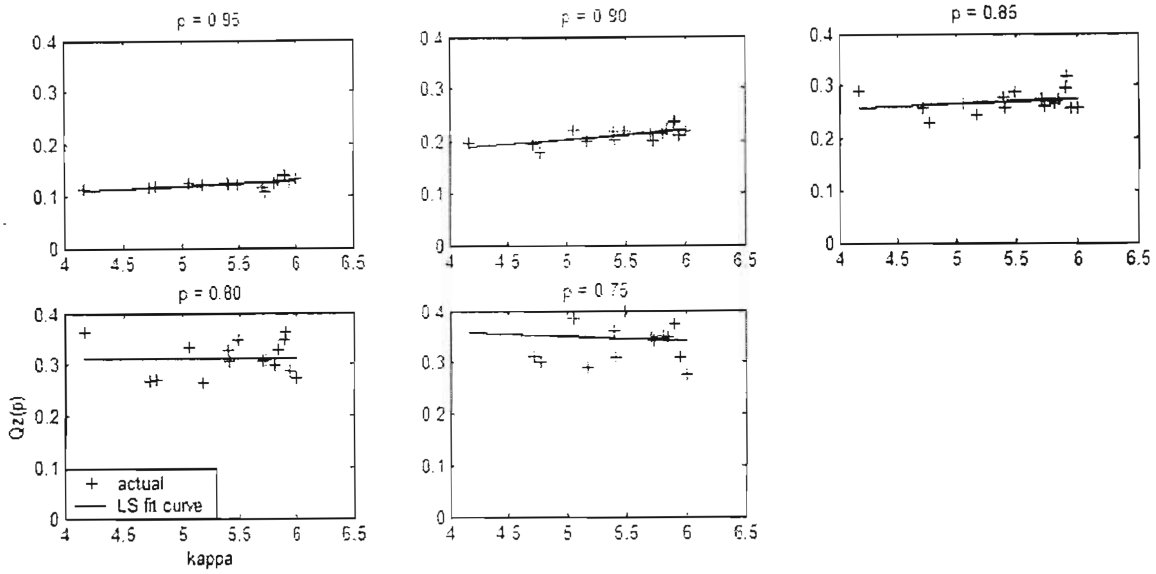


Figure 6-6: Correlation between Qz and kappa

The straight line fit is very apparent. Interestingly, our result is different from the original, shown in Chapter 2. This is on account of the different handling of the threshold to that of He and Mitra. The result above shows that Qz(ρ) is more strongly dependant on ρ that κ.

The least squares fit is implemented using the linear regression algorithm presented in [Chap98]. If $\underline{x}$ is the independent data vector, and $\underline{y}$ is the dependant data vector, each of length n, and the dataset is to be fitted to a straight line of equation:

$$y = a_0 x + a_1 y \tag{6.6}$$

by means of a least squares regression, then following equations may be used:

$$a_0 = \bar{y} - a_1 \bar{x} \tag{6.7}$$

and

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \cdot \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}\ . \tag{6.8}$$

This is implemented in Matlab as follows:

```
for i=1:numlevels,      %one regression for each value of p
   n = numimages;       %n, number of datapoints = number of images

   x = data3d(:,3,5);   %Qz
   y = data3d(:,i,2);   %kappa

   sigma_x  = sum ( x );
   sigma_y  = sum ( y );
   sigma_xy = sum ( x.*y );
   sigma_x2 = sum ( x.^2 );

   mean_x = sigma_x/numimages;
   mean_y = sigma_y/numimages;

   a(i,1) = (n*sigma_xy - sigma_x*sigma_y ) / (n*sigma_x2 - sigma_x^2);
   a(i,2) = mean_y - a(i,1)*mean_x;
end
```

Code Fragment 2: Linear Regression for Qz-kappa.

Hence the values of A_i and B_i required for the linear model (6.5), were found to be

$$A_i = \begin{bmatrix} 0.0094 \\ 0.0184 \\ 0.0086 \\ 0.0009 \\ -0.0093 \end{bmatrix}$$

and

$$B_i = \begin{bmatrix} 0.0709 \\ 0.1102 \\ 0.2231 \\ 0.3069 \\ 0.3961 \end{bmatrix}.$$

That A_i is at least an order of magnitude smaller than B_i, indicates the stronger dependency on ρ than κ, shown in Figure 6-6.

Using these values to generate an estimate of Qz using equation (6.5), allowed the following curves to be plotted. Each graph is a comparison between the 'first principles' Qz(ρ) and the fast estimate developed above.

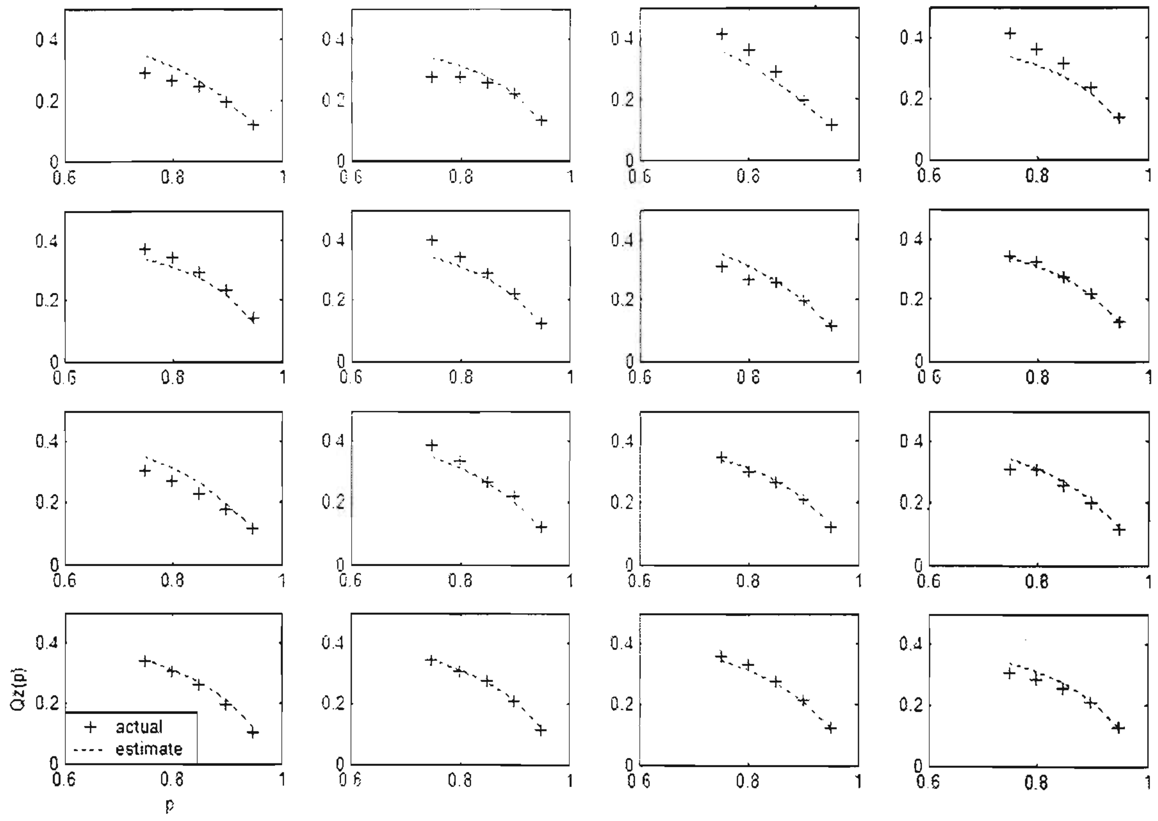


Figure 6-7: Comparison between Estimated and Actual Qz curves

This figures indicate how similar the Qz curves are, on average, as well as the closeness of the regressed curve to the actual curve, as expected from the theory.

6.2.3.1.3 Rate Function Regression

Having configured the fast estimators for Qz and Qnz, it remains to perform the regression necessary to calculate rate function:

$$R(\rho) = A(\rho)\cdot Q_{nz}(\rho) + B(\rho)\cdot Q_z(\rho) + C(\rho) \tag{6.9}$$

This now reduces to a calculation of $A(\rho)$, $B(\rho)$, and $C(\rho)$. [Chap98] shows that the function

$$y = a_0 + a_1x_1 + a_2x_2 \tag{6.10}$$

may be solved for $\underline{a}$ with a multiple linear regression. The solution for $\underline{a}$ is given by

$$\begin{bmatrix} n & \sum x_{1i} & \sum x_{2i} \\ \sum x_{1i} & \sum x_{1i}^2 & \sum x_{1i}x_{2i} \\ \sum x_{2i} & \sum x_{1i}x_{2i} & \sum x_{2i}^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum x_i \\ \sum x_{1i}y_i \\ \sum x_{2i}y_i \end{bmatrix}. \tag{6.11}$$

This is implemented in Matlab, using the following code

```
for i=1:numlevels,
   x1 = data3d (:,i,3); % Qz
   x2 = data3d (:,i,2); % Qnz
   y =  data3d (:,i,4); % Rate

   sigma_x1 = sum ( x1 );
   sigma_x2 = sum ( x2 );
   sigma_y  = sum ( y );

   sigma_x12 = sum ( x1.^2 );
   sigma_x22 = sum ( x2.^2 );
   sigma_y2  = sum ( y.^2 );

   sigma_x1x2 = sum ( x1.*x2 );
   sigma_x1y  = sum ( x1.*y );
   sigma_x2y  = sum ( x2.*y );

   A = [numimages sigma_x1 sigma_x2; sigma_x1 sigma_x12 sigma_x1x2; sigma_x2
sigma_x1x2 sigma_x22];
   y = [sigma_y; sigma_x1y; sigma_x2y];
   x(i,:) = (A\y)';
end
```

Code Fragment 3: Muliple Linear Regression

The values for the regressed variables are thus found to be

$$\begin{bmatrix} \underline{a_0} & \underline{a_1} & \underline{a_2} \end{bmatrix} = \begin{bmatrix} 0.1631 & -2.2986 & 0.43219 \\ 0.5958 & -5.4044 & 0.99439 \\ 0.0343 & -0.1520 & 0.00859 \\ 0.4257 & -1.3856 & 0.00616 \\ 3.4554 & -8.7220 & -0.32453 \end{bmatrix} \cdot 10^7 .$$

The multiplication by 10^7 is simply a result of the units used, the matrix converts between $\rho \in [0,1]$ and the rate, $R \in [0,6000]$. A final multiple linear regression is performed, as per [He01], to linearise this curve to match:

$$R(\rho) = \frac{1}{5}\sum_{i=1}^{5} R(\rho_i) + \theta\left(\rho - \frac{1}{5}\sum_{i=1}^{5}\rho_i\right), \tag{6.12}$$

where' θ is given by the regression formula:

$$\theta = \frac{\sum_{i=1}^{5}\rho_i \cdot \sum_{i=1}^{5} R(\rho_i) - 5\sum_{i=1}^{5} R(\rho_i)\rho_i}{5\sum_{i=1}^{5}\rho_i^2 - \left(\sum_{i=1}^{5}\rho_i\right)^2}. \tag{6.13}$$

This is expressed more fully in Chapter 2.

Using these values the rate of each image is estimated using equation(6.12), and compared with the rate generated by the actual algorithm. The results are given below.

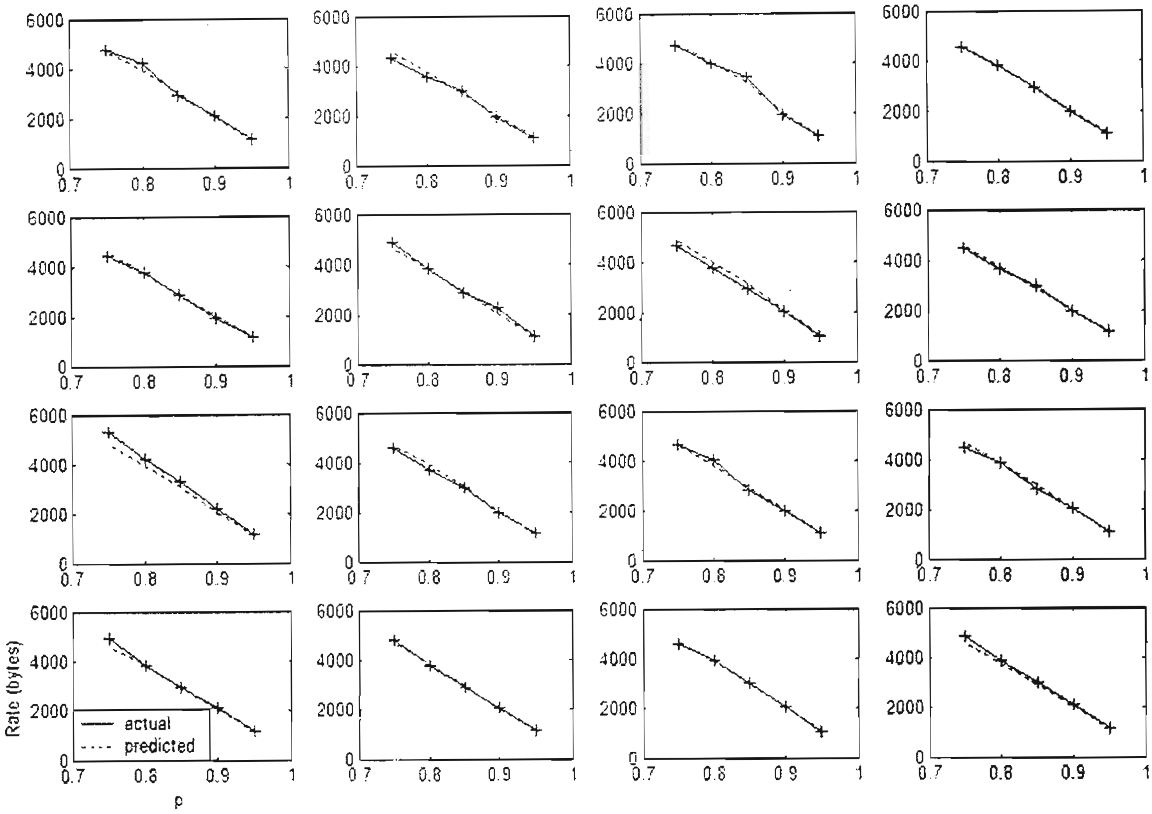


Figure 6-8: Rate Estimation

These results are a clear indication of the power of the ρ-domain method. The rate, as a function of ρ, is a very well defined and predictable function. This affords the method great advantage over all previous methods, which have defined the rate as a function of the quantization parameter. This consideration produces very diverse and unpredictable functions.

The code to implement the complete rate estimation is too long for inclusion here, and may be found on the accompanying CD.

6.2.3.1.4 Distortion Prediction

The next stage is an estimation of the distortion, also as a function of ρ. This is performed directly from the distribution of cuefficients.

The theory of estimating the distortion is given in Chapter 2. He and Mitra do not define their method, but state it that "D(q) may be directly computed from the distribution information." In [Jack03] a method is defined. A non-normalised, discrete distribution, $d(a)$, of the image coefficients is calculated where each $d(a)$ is the number of coefficients in the image whose absolute value equals a. This is possible due to the discrete nature of the data.

This function is performed in C with the following code.

```
/* This function generates the distribution of coefficients in the wavelet coefficient
image */
/* Image[][] is the 2D coefficient map, and y_dim and x_dim are the dimensions of the
image */

void calc_distribution(int** Image, int y_dim, int x_dim, float* distribution) {
        int x, y,n;

        /* First clean the distribution */
        for (x = 0; x < 256; x++)          {
                        distribution[x] = 0;
        }

/*iterate through the image and calculate the distribution */
        for (y=0; y<y_dim; y++)
                for (x=0; x<x_dim; x++)          {
                                if ( abs(Image[y][x]) >= 255 ) /* hard limit at 255 */
                                        distribution[255]++;
                                else
                                        (distribution[ abs(Image[y][x]) ])++;
                }
        }

}
```

Code Fragment 4: Distribution Calculation

The behaviour of SPIHT may now be utilised to determine an estimate of the distribution of the output coefficients. At the end of a significance pass, all the coefficients within the image that have magnitude below the threshold, will effectively have been quantized to zero. This allows ρ to be easily calculated:

$$\rho(\Delta) = \frac{1}{N}\int_{0}^{\Delta} d(a)da\,. \tag{6.14}$$

Here N is the total number of coefficients in the image and Δ a SPIHT threshold value.

In addition, the MSE of the compressed image may be calculated from $d(a)$ and Δ using:

$$D(\Delta) \cong \frac{1}{N}\left(\int_{0}^{\Delta} a^{2} d(a)da + \int_{\Delta}^{\infty}\left(\frac{\Delta}{2.5}\right)^{2} d(a)da\right). \tag{6.15}$$

The first term accounts for the distortion introduced as a result of zero-quantising insignificant coefficients, and the second term approximates the distortion introduced by truncating the insignificant bitplanes in the significant coefficients, as discussed in the calculation of Qz and

The code fragment shows an iterative approach to solve for D(ρ). Equation (6.14) is initially used to find the Δ corresponding to the desired ρ. Then this value of Δ is used in equation (6.15) , to calculate the distortion at this value of ρ.

This predicted distortion is compared to the actual value of distortion produced by the algorithm, as shown in Figure 6-9.

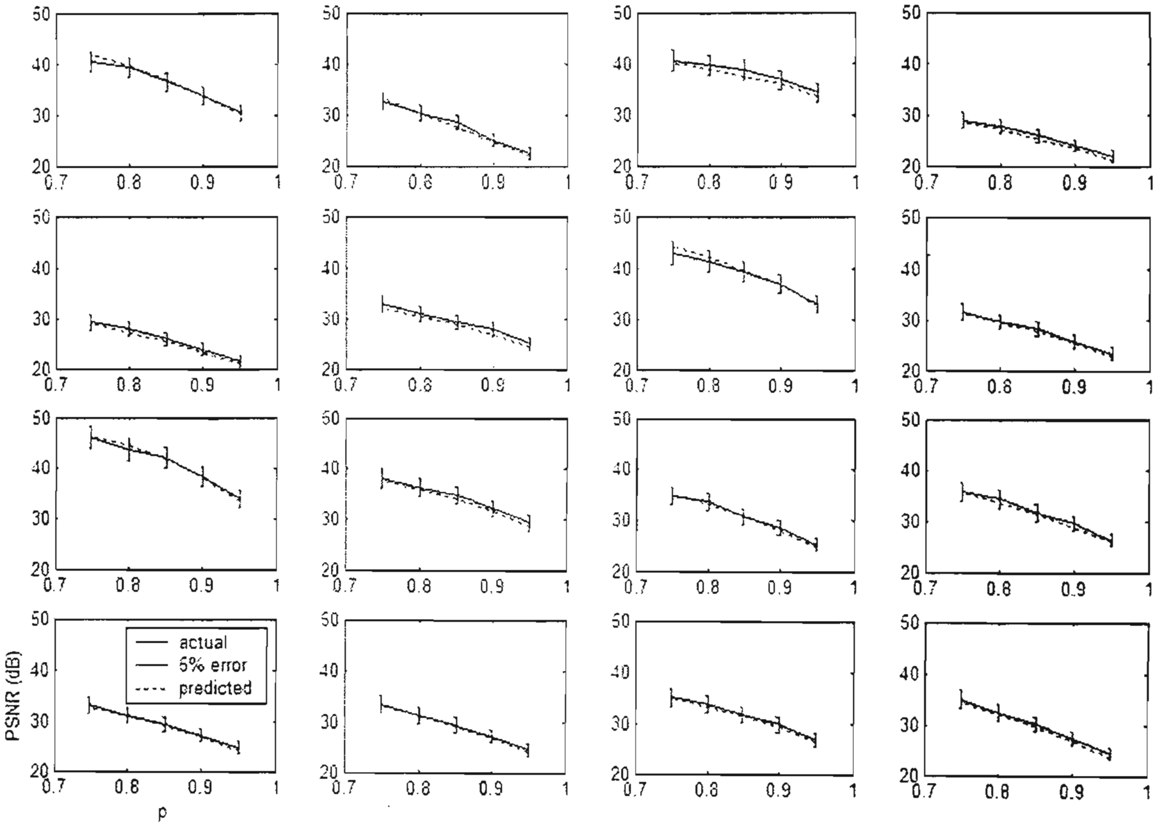


Figure 6-9: Comparison of Estimated with Actual Distortion

These curves show PSNR which is quality, as opposed to MSE distortion, but is simply given by equation 3.2 in Chapter 3. PSNR is used in preference to MSE as it produces more linear curves, which facilitates display. Internally however, the algorithm uses MSE.

These results show that although the distortion behaviour is diverse across the image set, this fast estimate is accurate to within 5% in all cases.

6.2.3.1.5 Combined RD Estimate

Finally as both the rate, R(ρ), and distortion, D(ρ), are parameterised by ρ, the curve of D(R) may be easily found. The curves below compare the estimated D(R) at the five values of ρ, with the actual D(R) found by running the SPIHT algorithm multiple times (ie the Operational D(R) curve).

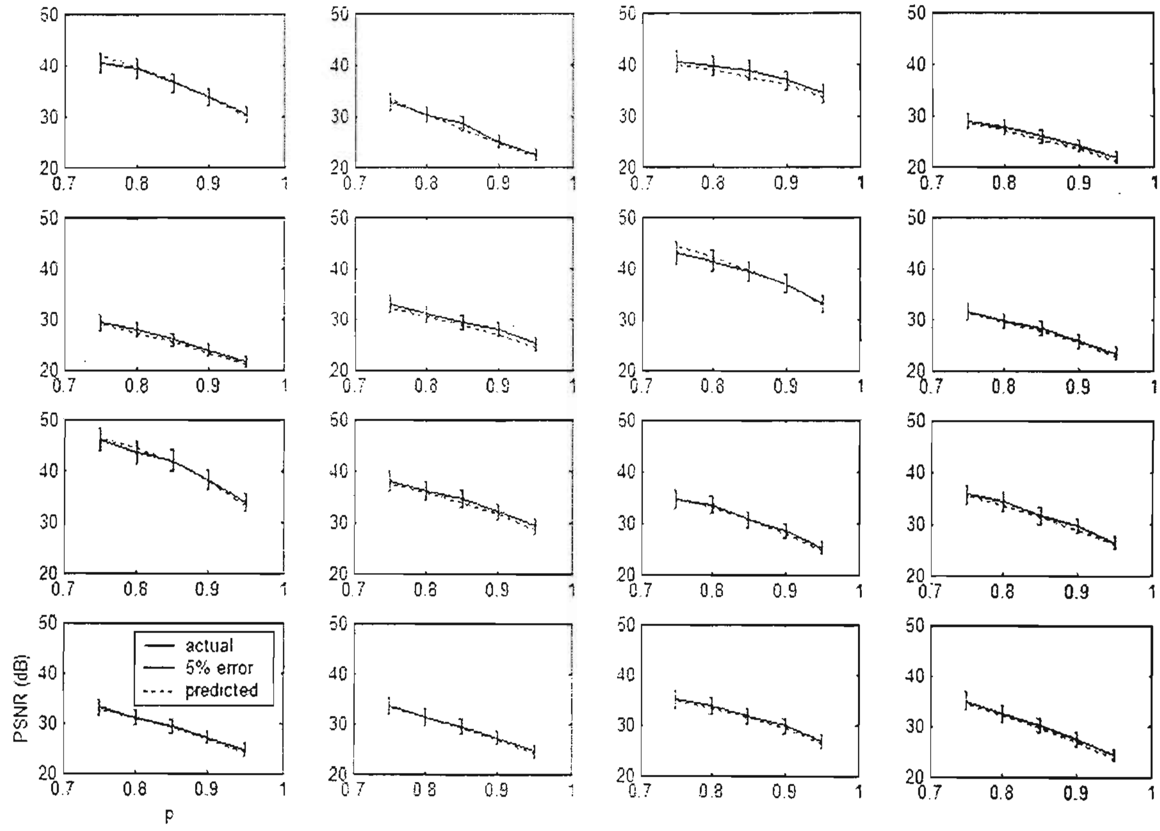


Figure 6-10: Estimated and Actual PSNR-Rate Curves

Again the curves shown are in terms of PSNR, not MSE. These curves illustrate the great accuracy of the ρ-domain method, successfully capturing the different behaviour of the images, using a very fast implementation.

6.2.3.1.6 Testing Set

In order to test the ability of the algorithm to predict RD behaviour, a set of new, unseen, images was passed through the RD estimator, and the compressor. These images are shown in Figure 6-11.

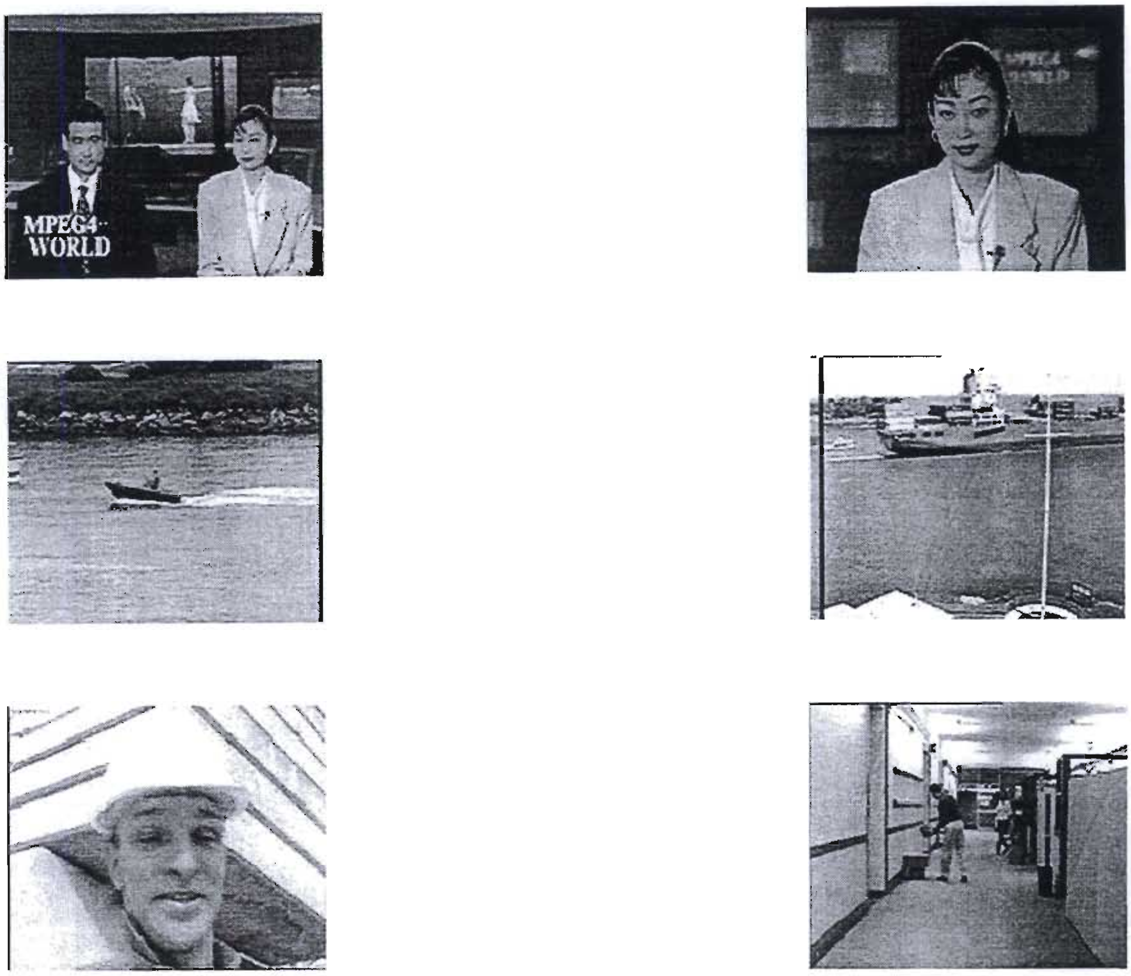


Figure 6-11: Image Test Set

This test set was chosen from the set of standard MPEG video test sequences, choosing a variety of natural scenes and different levels of zoom. A comparison of the RD results is given below:

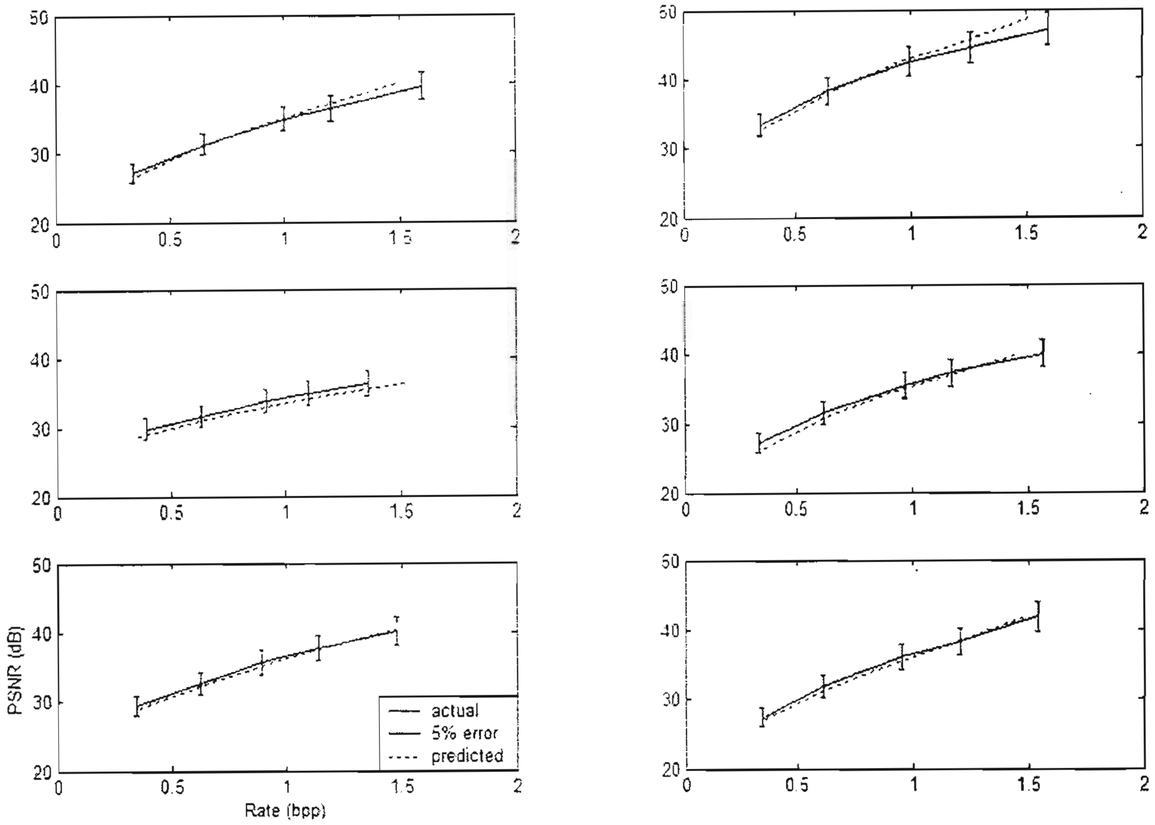


Figure 6-12: Comparison between Actual and Estimated RD Behahaviour (Test Set)

Clearly the algorithm is able to estimate the RD behaviour of this unseen dataset. This experiment confirms the operation of our implementation of He and Mitra's algorithm.

6.2.3.2 RD Estimation of Difference Frame Tiles

Having produced an algorithm capable of performing RD estimation on general still images, it was required to test the performance on difference frame tiles. This section will discuss this step. As the procedure is identical to the one described for still images, only the differences will be discussed here.

6.2.3.2.1 Training Set

In order to provide the configuration data for the regression, the compression algorithm was run on the 'Hallmonitor' sequence, and difference frame 100 extracted. This difference frame was segmented into 16 tiles, each of which was treated separately by the RD estimator. This difference frame is in Figure 6-13.

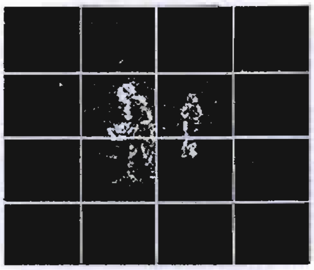


Figure 6-13: Difference Frame 100 of Hallmonitor Sequence

This frame was chosen for the diversity of the tiles, most of the tiles are very low energy, but the four central tiles display increasing energy. This diversity is important, in order for the regressions to provide general results.

6.2.3.2.2 Target Rate

Before performing the configuration it is necessary to choose the range of ρ over which the algorithm is to perform. As this target application is very low bitrate video the highest value of $\rho = 0.99$ was chosen. In addition the strong spatial clustering of significant coefficients motivates that some tiles receive many bits, thus a lower value of $\rho = 0.8$ was chosen. This choice allows the bit allocation stage to allocate bits liberally where the variance is high, and sparsely in low variance region, thus allowing good RD performance.

6.2.3.2.3 Rate Estimation

The rate estimation configuration exercise was performed as before, generating a new set of coefficients. The following curves show a comparison between the estimated and actual rate.

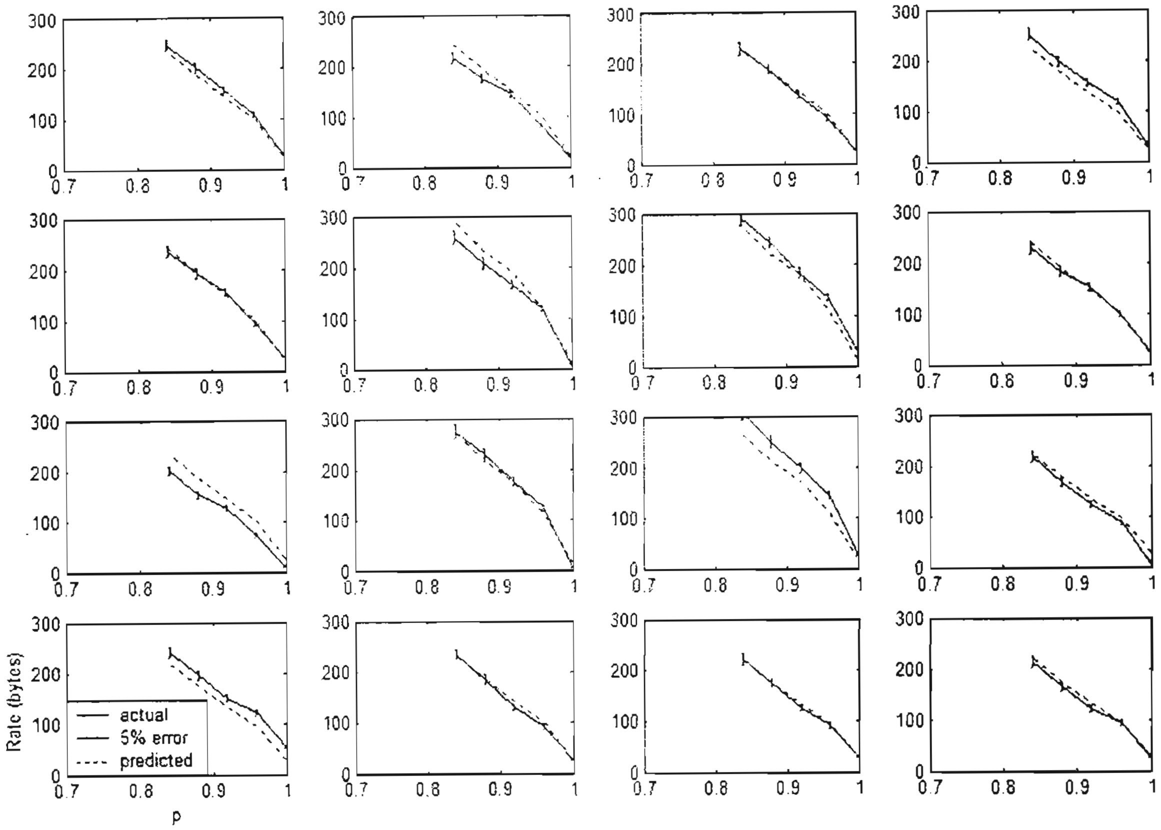


Figure 6-14: Comparison between Estimated and Actual Rate

Having observed the rapid decrease of the actual rate curve above $\rho = 0.95$ in Figure 6-14 it was decided to abandon the linear rate regulation, as the rate curve is no longer approximately linear. This outcome illustrates the statistical difference between normal and difference residual frames. Figure 6-8 shows the rate behaviour of normal images, while Figure 6-14 shows the rate behaviour for difference images. Due to the relatively large number of zero and near-zero coefficients in difference frames, the behaviour of the coding algorithm is expected to change at high values of ρ and this behaviour is clearly shown in Figure 6-14.

6.2.3.2.4 Distortion Estimation

The distortion estimator was configured as before, and the following presents its performance.

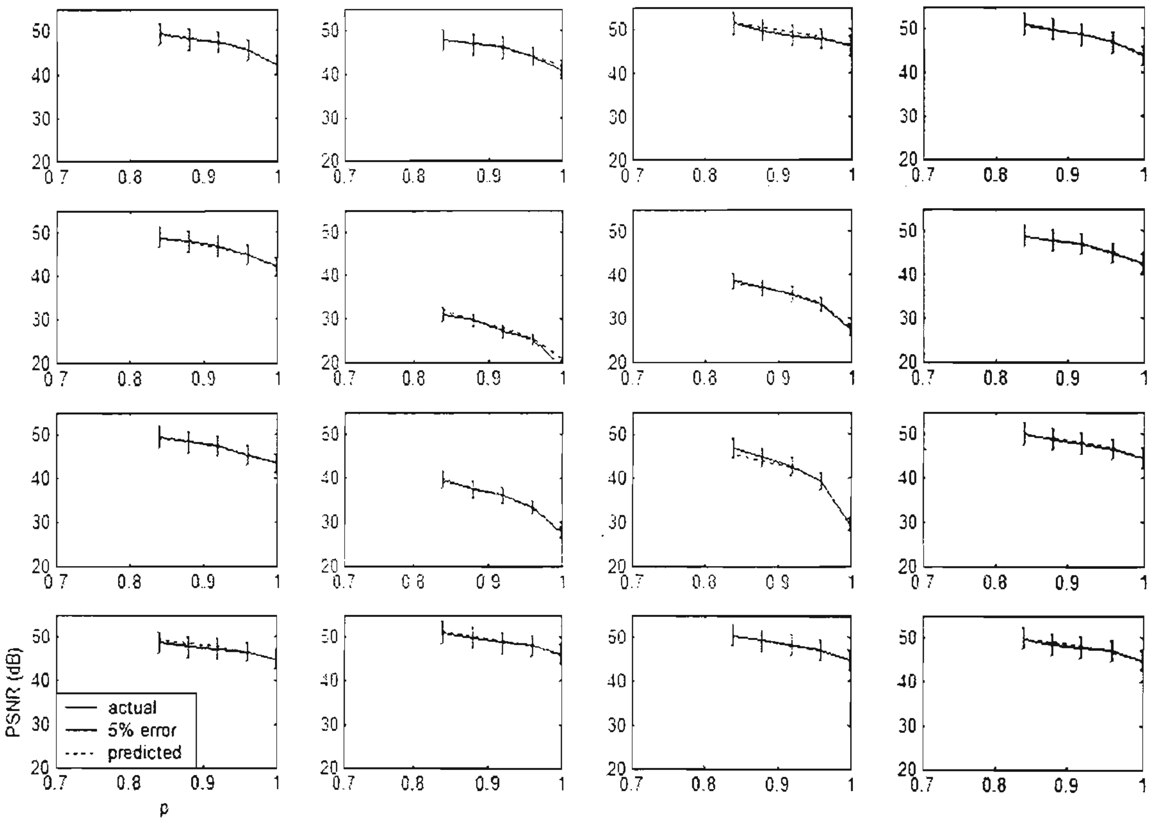


Figure 6-15: Comparison of Estimated and Actual Distortion

These results show the range of distortion encountered over a single difference frame, further evidence of the utility of source partitioning. In addition the good performance of the distortion estimator is shown.

6.2.3.2.5 RD Estimation

Combining the rate and distortion estimation as before, the following curves are produced.

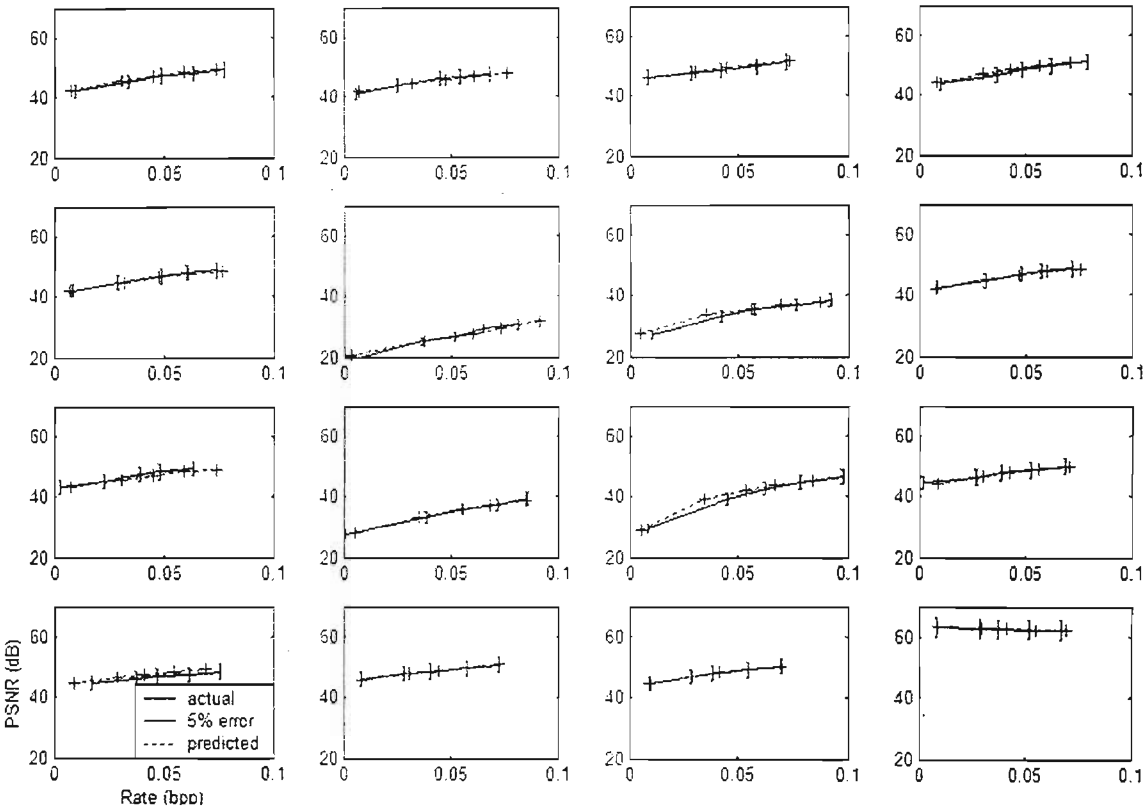


Figure 6-16: Difference Frame RD Estimation Performance

These curves immediately show the varied behaviour between the tiles. As intuitively expected, the PSNR possible at a given rate is lower over the central 4 tiles, as the signal energy is concentrated there.

6.2.3.2.6 Generalised Performance

In order to test the performance of the algorithm, several other frames were input. The result for difference frame 100 of the 'Foreman' sequence is shown in Figure 6-17.

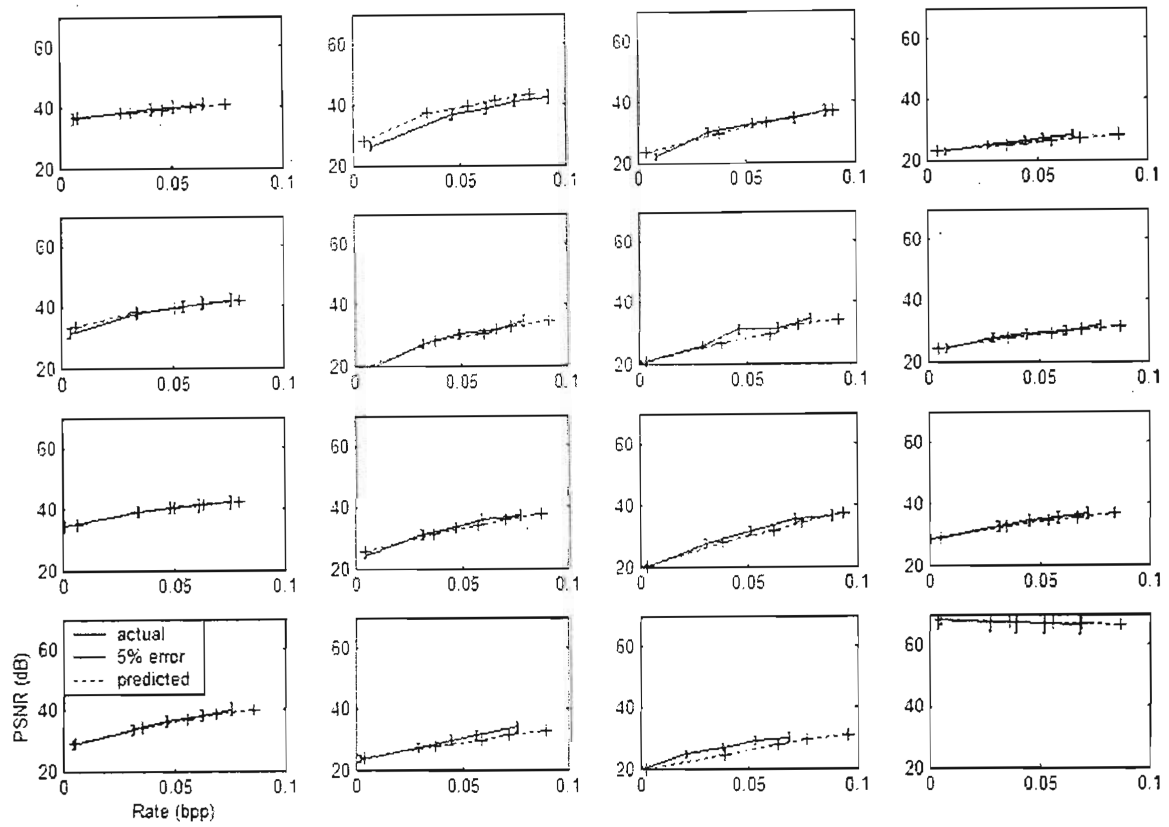


Figure 6-17: RD Estimation of Foreman Frame #100

Although omitted on space considerations, the other standard sequences were tested, and produced similarly good results.

6.2.3.2.7 Conclusion of RD Estimation Stage

Although the performance of the RD estimation for difference frames is not as good as for natural images, it was deemed sufficient for the algorithm. Several changes from the original He and Mitra algorithm were made in order to increase the performance of the algorithm.

6.2.4 Optimal Bit Allocation Stage

Once reliable RD estimation has been performed it remains to use this information to distribute the available bits between the tiles in an optimal fashion.

The problem is considered in two stages;

- For a given λ, divide the bits between the tiles using the Lagrange method outlined in Section 2.2.3.
- Use a bisection search to match the value of λ used in the Lagrange stage with the desired bit-rate. Iterate through step 1, until the desired rate is met.

This technique was adapted from [Ram93] where a similar strategy is used to drive an optimal tree pruning algorithm for a best wavelet packet bases algorithm.

6.2.4.1 Lagrange Sub-stage

For a particular difference residual frame, the set of R(D) curves for each tile may look like the following set, taken for 'Foreman' frame number 100.

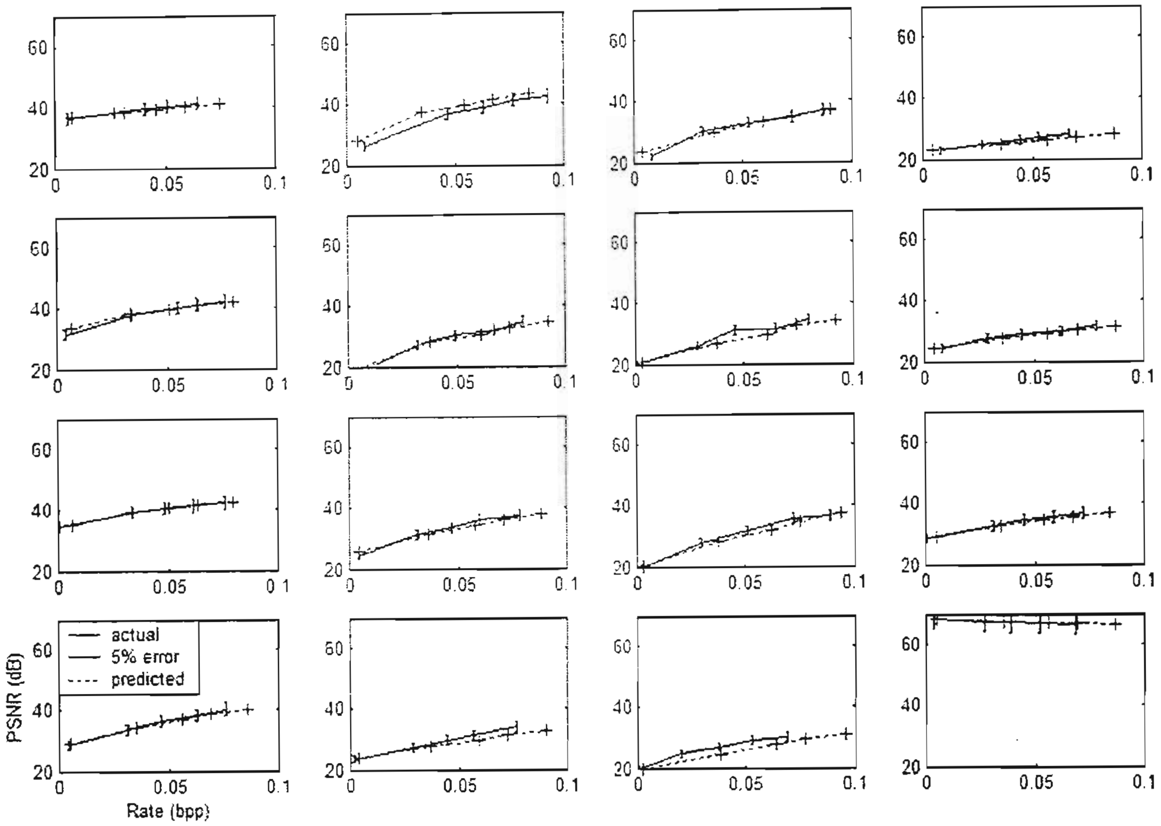


Figure 6-18: RD Estimation of Foreman Difference Frame #100

Although these curves show the PSNR, the algorithm works internally with MSE, as the MSE metric is additive across the image (with appropriate scaling of the final result).

As Section 2.2.2 proves in detail, finding the state where each curve operates at the same gradient, λ, will provide the minimum possible distortion for a particular rate. Thus, this stage takes as input a desired value of λ and searches the possible states such that each tile operates as closely to this gradient as possible. As this is a rather trivial exercise it will not be detailed here.

6.2.4.2 Bisection Sub-stage

The bisection has not been described previously and will be detailed here in algorithmic form. This discussion is based heavily on [Ram93]. As the MSE curve is used the gradient of any point on the curve will be negative. however in the following discussion, only the magnitude of the gradient will be used.

The rate curve is guaranteed to have a concave shape [Ram93]. The concavity allows a bisection search to find the value of λ corresponding to the rate budget.

1) **Initialise:** Choose an upper and lower value for λ, λ_u and λ_l, such that

$$\sum_i R_i^*(\lambda_u) \le R_{budget} \le \sum_i R_i^*(\lambda_l)\,. \qquad (6.16)$$

That is the optimal bit distribution between the tiles for λ_u, will produce a rate $\sum R_i^*(\lambda_u)$ that is less than the total rate budget for the frame, R_{budget}, which in turn is less than the rate for the optimal distribution for the gradient λ_l.The values of λ_u and $\lambda_l$are initially chosen to be $\lambda_l = 0$ and $\lambda_u = 10^{10}$ to ensure this state.

2) **Find Optimal Bit Distribution:** Call the Langrange routine of Section 6.2.4.1 to calculate the optimal bit distribution for λ_u and λ_l.

3) **Bisect:** Find the next value of λ, λ_n, by finding the gradient between the upper and lower points on the R(D) curve.

$$\lambda_n = \left|\frac{\sum_i D_i^*(\lambda_l) - \sum_i D_i^*(\lambda_u)}{\sum_i R_i^*(\lambda_l) - \sum_i R_i^*(\lambda_u)}\right|$$

This operation is the bisection that gives the algorithm its name.

4) **Bounds Adjustment:** One of the bounds is now replaced such that $\sum_i R_i^*(\lambda_u)$ and $\sum_i R_i^*(\lambda_l)$ are centred more tightly around R_{budget}.This is easily achieved:

if $\left(\sum_i R_i^*(\lambda_n) = R_{budget}\right)$ then the desired bitrate has been achieved, and is set to: $\lambda^* = \lambda_u$.
if $\left(\sum_i R_i^*(\lambda_n) > R_{budget}\right)$ then the lower bound must be tightened: $\lambda_l \leftarrow \lambda_n$,
else the upper bound must be tightened: $\lambda_u \leftarrow \lambda_n$

5) **Reiteration:** Repeat from step 2, unless the number of iterations has exceeded a threshold (included to limit the maximum time spent in the algorithm).

This is a completely standard bisection root finding method, and is guaranteed to converge to the solution due to the concavity of the rate distortion function.

6.2.4.3 Conclusion of Optimal Bit Allocation

This stage finds the point on the R(D) curve of each tile that corresponds to the lowest overall distortion. This operating point is used to control the output stream length of each tile during the SPIHT coding stage.

6.2.5 Spatial Coding Stage

Each tile within the frame is encoded separately using the SPIHT coder, to produce the length specified by the optimal bit allocation routine. This code was implemented directly from [Said96] with few significant changes. Thus, although it represents the most significant portion of the programming exercise, it warrants no further comment here.

6.2.6 Entropy Coding Stage

The entropy coding stage is also completely standard. The algorithm was not written by the author, but was taken almost verbatim from [McIn02]. In that work, it is also used to code the output of a SPIHT coder. Implementation details of this code are discussed in that work.

6.3 Conclusion

This chapter has proposed a new video compression algorithm. This algorithm is based mainly on the source modelling exercise of Chapter 5, as well as many of the ideas and algorithms of Chapters 3 and 4.

The main imperative of this project is the production of a low complexity video coder, for eventual deployment in a mobile device. This intended application places severe restrictions on the sophistication of the coding algorithm due the meagre computational resources available on mobile devices. This has motivated each decision taken during the algorithmic design.

The primary complexity saving is through abandoning block ME/MC, which accounts for over 80% of the execution time of recent video coding standards. Unfortunately, the standards include block ME/MC for its excellent RD advantages and our exclusion of this algorithm is expected to incur severe RD costs.

Chapter 5 shows that partitioned coding of difference frames offers a possible low complexity temporal decorrelation strategy. The chapter couples this temporal strategy with appropriate spatial and entropy coding algorithms. The underlying assumption is that the RD performance losses required to produce a sufficiently simple algorithm may be recouped, at least in part, through clearly directed spatial and entropy coding. Furthermore these algorithms are dramatically faster in execution than block ME/MC, thus this strategy is an efficient one.

This chapter initially discusses the block level design of the algorithm. Then increasing levels of detail are added to add substance to the discussion, this mirrors the actual design process. First the choice of algorithm for each block in the system is discussed and justified from the literature reviews of Chapters 3 and 4. Thereafter the main design element of this project; the configuration of the ρ-domain RD estimator of wavelet difference frames, is discussed in detail. This algorithm was implemented from the [He01] and tested on a still image set. Having confirmed the algorithm functionality, it was modified and reconfigured for use in RD estimation of wavelet difference frames. As the discussion shows, significant effort was required to configure the various regression parameters of this algorithm.

Other elements of the project, such as the optimal bit allocation, SPIHT and entropy coding routines are more mundane and their discussion is intentionally brief. The referenced literature provides an abundance of detail on these algorithms, which do not justify repetition here.

Most of the programming detail has been omitted due to space constraints. However, the entire source code for the project is on the attached CD. This code is well commented, and self-explanatory.

This chapter has shown how at each stage the computational load of the implemented algorithm has been of primary concern. Difference frame coding was chosen to gain complexity savings and sophisticated spatial and entropy coding have been employed to ameliorate the concomitant RD degradation. Chapter 7 to follow will test the success of this strategy.

Chapter 7 - Results and Discussion

This chapter presents a comprehensive performance evaluation of the video compression scheme detailed in Chapter 6. There are two elements to the system performance; the complexity and the RD performance. The primacy of the criterion of low system complexity has been laboured in the preceding chapters and it is imperative to determine if the proposed system in fact succeeds in meeting this requirement. Thus, tests of the overall system complexity, as well as the complexity of each sub-system, are made below. The trade-off made for low complexity is RD performance and it is equally important to determine whether the system presents acceptable RD behaviour. Thus the purpose of this chapter is to analyse the complexity and RD behaviour of the proposed scheme, and determine the suitability of this system to our application.

Fortunately the application is specific, which allows clear requirements to be stated. Mobile devices are limited in most respects, their computational limits already having been discussed. Here, the available bandwidth and display resolution are of interest in determining acceptable performance. Although the introduction in Chapter 1 has indicated that commercial networks are rapidly increasing their bandwidth, our particular application is to a 19.6 kbps channel, which is extremely limited. Thus our first goal is just to transmit video over this channel, regardless of resolution. The second matter of display resolution is also relevant. For instance an industry leading palm-top computer, the HP iPAQ 5555, has a screen resolution of 240 x 320 and a 16-bit colour display. On a device such as this, large images are not really useful. Furthermore, very high image quality is not necessary, as most devices cannot display them. Thus our tests will focus on low bit rate coding of small images.

Section 7.1 will present the experimental environment in which the tests were undertaken. The computing hardware and software environment is clearly specified, as is the test data set. This section is included to explain the results and allow reproducibility.

Section 7.2 will compare the proposed algorithm to the standard video coders; H.263+ and MPEG-4. These algorithms are the benchmark of performance, representing the current state of the art. Any proposed algorithm must present performance at least equal to these algorithms to be considered competitive. Furthermore, as these are algorithms published by standards-setting bodies, reference software is available and rigorous testing may be done. This discussion will be structured by test sequence; where for each sequence the computational and RD performance of the algorithms will be presented together and compared. This section highlights the strengths and weaknesses of our algorithm when compared to the existing methods.

Section 7.3 will compare the proposed algorithm to the various wavelet based video schemes proposed in the literature. As the source code for these algorithms is unavailable, rigorous testing is difficult. Thus we have attempted to reproduce, as nearly as possible, the experiments described and compare the results of our algorithm with the published results. This discussion will be structured by literature algorithm. This discussion places our algorithm within the context current research.

Section 7.4 will analyse the various components of the proposed algorithm in more detail. The complexity and RD characteristics of each major stage in the proposed system are separately examined. The purpose of this is mainly to test the assumptions made during the algorithmic design of Chapters 5 and 6. Aspects such as coding of difference frames, and spatial bit allocation will be tested to determine their RD and complexity implications.

Finally Section 7.5 will present a unified discussion of these results.

From the previous chapters it is expected that our algorithm will be computationally simpler than other schemes, due to abandoning ME/MC. In addition it is expected that due to the spatial bit allocation and use of the wavelet transform, good visual quality will be produced.

7.1 Experimental Method

This section will detail the experimental environment in which all the tests of the chapter are conducted. Choices such test data set and algorithmic configuration will be justified.

7.1.1 Platform

The computational platform used to perform all of these tests is an Intel Pentium 4, 1.6GHz PC, with 512MB of RAM, running Windows XP 2002 Professional SP1.

All the software implementations have been compiled under Microsoft Visual C++ 6.0, and are executed from within this environment.

The H.263+ algorithm [H263] was obtained from the University of British Columbia. This software is based on the Telenor codec, and is compliant with H.263 TMN8 [H263,2]. It is run with the following configuration, in all cases:

Quantization Parameter	13
Motion Vector Mode	H.263+
Syntax Based Arithmetic Coding	Enabled
Advanced Prediction Mode	Enabled
Deblocking Filter	Enabled
Advanced Intra-coding Method	Enabled
Reference Picture Selection Mode	Enabled

Table 7.1: H.263+ Configuration Options

This configuration does not use unlimited motion vector search range, in order to limit the computational overhead of the algorithm, this is a common practice. Furthermore, PB frames are not enabled, for the same reason, as well as incompatibility with other options. This configuration may be considered to be low complexity, hence a good comparison for our algorithm.

The MPEG-4 algorithm is run with the default options.

The results for the standards based methods presented below differ from the results presented in Chapter 4, in most cases. Specifically the H.263+ results are consistently below published results. This is on account of differences in the implementation, vendor specific optimisations and different configuration options. Furthermore, the decision not to use PB frames or unrestricted motion vectors in order to try limit the computational complexity, is at the cost of RD performance. It was decided to use the above configuration in all cases, rather than modify the options for each test sequence to match other published results. This enables a standard comparison to be made for all sequences and rates.

7.1.2 Computational Complexity

The computational complexity of a video compression algorithm is a difficult quantity to derive analytically due to the size and source dependency of these algorithms. Thus it is common practice to compare the complexity of algorithms by executing them on a common platform and measuring the execution time. Although this method is subject to several hazards such as machine dependant effects, code implementation variations and the impact of possible background processing, it does give an indication of the underlying complexity.

Each algorithm was profiled using the Microsoft Profiler, which is a standard accessory in Microsoft Visual Studio. The Profiler returns the amount of time spent in each function of an algorithm.

7.1.3 Test Data

The intended application of this algorithm is mobile, handheld devices, thus low resolution, low to medium frame rate video is appropriate test data. The data used is a set of standard MPEG test sequences. Each frame is raw video QCIF format; 176x144 pixels, 8bit grey-scale, at 10 fps, in 4:2:0 (sometimes called 4:1:1) format. The video characteristics of each sequence will be discussed with its results. The original video is 30 fps, but unless otherwise stated we have used 10 fps video in testing, by only encoding every third frame.

Although the results are presented at specific rates, such as 20 kbps, the codecs are not always able to produce this exact rate. Unless otherwise stated the rate is accurate to 1 kbps. This lack of reporting precision is to aid readability. The accurate test result data is available on the accompanying CD for reference.

7.2 Comparison to Video Compression Standards

This section will code various test sequences using H.263+, MPEG-4 and the proposed algorithm. In each case the complexity and RD behaviour is measured and compared. The visual quality is compared both numerically, and a sample frame (#100 is all cases) is displayed to indicate perceived visual quality. This structure allows the algorithms to be compared on equal ground, and the diversity of test sequences considered highlights the strengths and weaknesses of each algorithm.

As mentioned in Chapter 5, standard implementations of H.263+[H263] (variously referred here to as H.263 and H.263+, although H.263+ is the correct term) and MPEG-2[MPEG2] and MPEG-4 [MPEG4] have been obtained.

MPEG-2 will only be considered once, for completeness. MPEG-2 is designed for video broadcast systems, and can only output a minimum frame-rate of 24 fps, thus it is unable to reach the low bit-rates of interest in this project. Furthermore it is far less sophisticated than the other algorithms, and subsequently is unable to compete on visual quality.

7.2.1 Akiyo

7.2.1.1 Characteristics

The Akiyo test sequence is a head and shoulders sequence of a news reader. There is no global motion, limited local motion, and no background motion. An example image is shown in Figure 7-1.

Figure 7-1: Original Akiyo Frame #100

7.2.1.2 Computational Complexity

	H.263+		MPEG-4		Proposed
Rate(kbps)	Time(ms)	ME/MC(%)	Time(ms)	ME/MC(%)	Time(ms)
15	26137	85.33	X	X	2792
20	38667	83.47	X	X	
25	38814	82.94	X	X	2889
40	39128	86.08	33615	89.30	2968
50	39138	84.70	X	X	6209

Table 7.2: Complexity Results: Akiyo

The table compares the two leading standard techniques with the proposed algorithm. For each algorithm the first column displays the total time spent in coding the sequence, and the second column gives the percentage of that time spent in ME/MC algorithms.

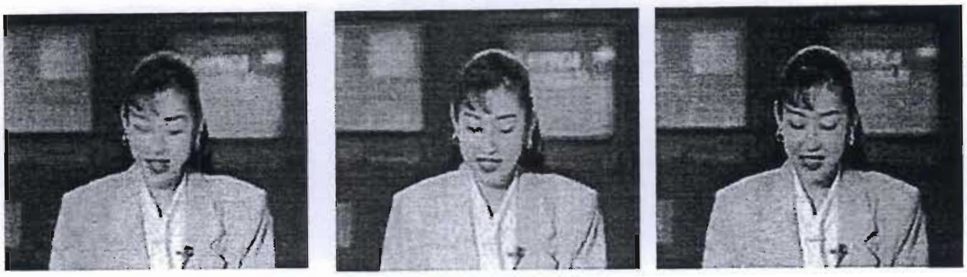
The publicly available reference software for MPEG-4 allows no user control over the bit rate, thus the compressed stream is only available at one possible rate, hence the missing data in the table above, and those to follow. This defect is noted in the manual that accompanies the reference software. The MPEG-4 codec which was run, did not include the object based coding that was discussed in Section 4.2.3, as this is not functional at this time. The video was coded with a block based ME/MC algorithm. We believe that due to great similarity between the MPEG-4 and H.263+ algorithms, the MPEG-4 algorithm will behave similarly to the H.263+ algorithm for the data rates not included on the table.

7.2.1.3 RD Performance

The RD performance is given below.

	H.263+		MPEG-4		Proposed	
Rate(kbps)	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps
20	31.38	9.8	X	X	28.63	10
25	32.16	10.6	X	X	32.60	10
40	34.41	11.6	34.43	10	36.18	10
50	35.18	11.5	X	X	38.00	10

Table 7.3: RD Performance: Akiyo



H.263 MPEG-4 Proposed

Figure 7-2: Example Output Images: Akiyo, frame 100, 40 kbps

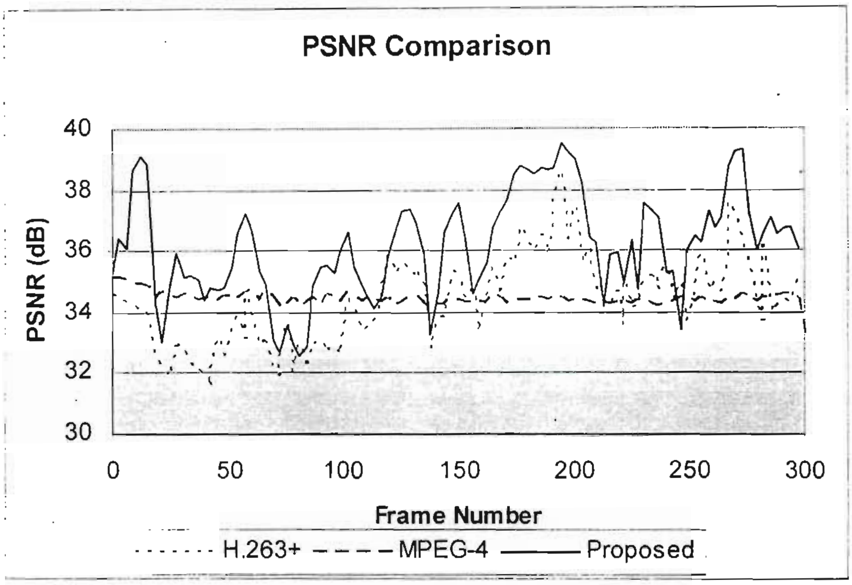


Figure 7-3: Frame by Frame PSNR: Akiyo (40 kbps)

The numerical and visual advantage of the proposed scheme is apparent.

7.2.1.4 Discussion

The first observation from this data is that the standard methods dedicate the bulk of their processing time to ME/MC. This is due to the exhaustive block matching searches that these algorithms engage in, in order to estimate motion. This effect is noted in Chapter 5 and is the fundamental motivation for our algorithm.

The second observation is that the proposed algorithm is an order of magnitude faster than the standard methods. This may be on account of implementation specifics, but the previous discussion of Chapters 5 and 6, motivates that this is not the case.

Furthermore the visual quality is perceptually and measurably superior to the standard algorithms. Figure 7-3 shows a weakness of the proposed algorithm; the large inter-frame PSNR fluctuations. This will be discussed in the general discussion in Section 7.5.

7.2.2 Akiyo MPEG-2

For completeness the MPEG-2 algorithm is tested in this section.

	MPEG-2	
Rate(kbps)	Time(ms)	ME/MC(%)
40	25112	40.33
60	25574	40.11
100	25239	40.33

Table 7.4: Complexity Results: Akiyo, MPEG-2

Table 7.4 indicates the performance of MPEG-2. .The reference software obtained is unable to produce data below 40 kbps. It is noted that this algorithm is less heavily reliant on ME/MC than the other algorithms.

The RD performance is given below.

	MPEG-2	
Rate(kbps)	PSNR(ms)	fps
40	28.60	30
60	37.57	30
100	43.97	30

Table 7.5: PSNR Results: Akiyo, MPEG-2

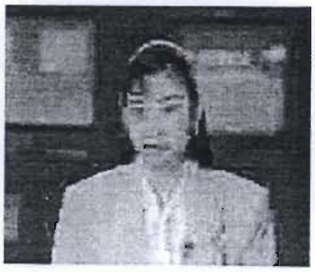


Figure 7-4: Sample Image: Akiyo, MPEG-2, frame 100, 40 kbps

As noted previously the frame rate for MPEG-2 is 30 fps and can only be reduced to 24 fps. This restriction limits the RD performance, and hence renders the algorithm unsuitable for our application.

7.2.3 Hallmonitor

7.2.3.1 Characteristics

The Hallmonitor test sequence is a surveillance camera scene. There is no global motion, large local motion with appearing and disappearing objects, and no background motion. An example image is shown in Figure 7-5.

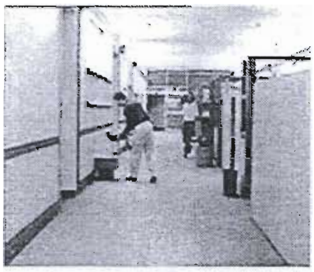


Figure 7-5: Original Hallmonitor Frame #100

7.2.3.2 Computational Complexity

	H.263+		MPEG-4		Proposed
Rate(kbps)	Time(ms)	ME/MC(%)	Time(ms)	ME/MC(%)	Time(ms)
15	15596	64.12	X	X	2701
20	38450	82.95	X	X	2725
25	39543	82.82	X	X	2865
40	39323	85.02	39010	82.27	3344
50	42120	79.79	X	X	6329

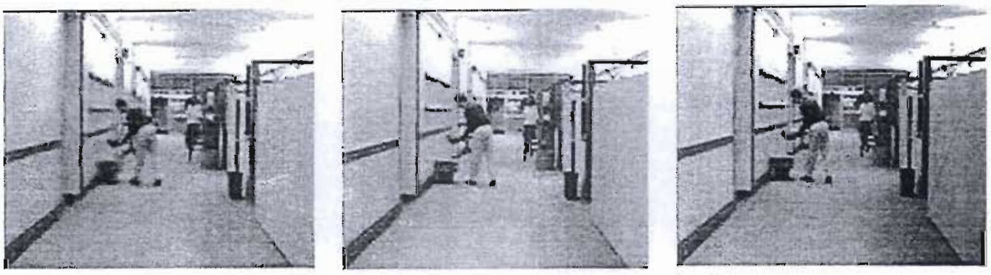
Table 7.6: Complexity Results: Hallmonitor

7.2.3.3 RD Performance

The RD performance is given below.

	H.263+		MPEG-4		Proposed	
Rate(kbps)	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps
20	29.68	9.8	X	X	28.86	10
25	29.95	10.1	X	X	29.87	10
40	31.53	11.2	33.23	10	33.84	10
50	32.79	11.3	X	X	34.90	10

Table 7.7: RD Performance: Hallmonitor



H.263 MPEG-4 Proposed

Figure 7-6: Example Output Images: Hallmonitor, Frame 100, 40 kbps

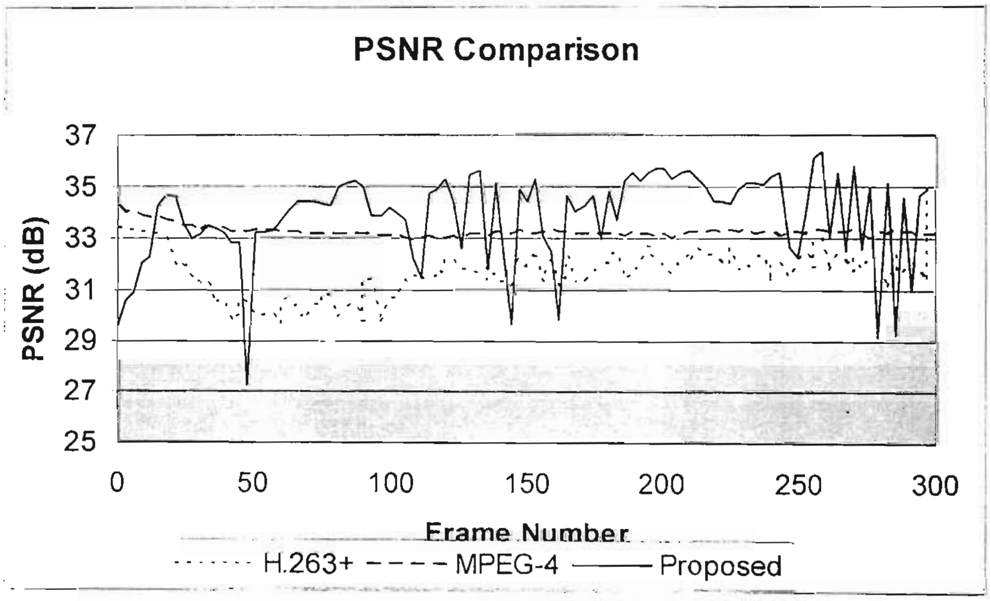


Figure 7-7: Frame by Frame PSNR: Hallmonitor (40 kbps)

7.2.3.4 Discussion

Although the proposed algorithm performs well again, the lack of interframe rate control is apparent in the rapidly fluctuating PSNR. The movement of the men in this sequence is much less smooth than the movement of Akiyo. Thus for those difference frames with much information, the algorithm is unable to allocate more bits than for those with little information.

As the movement is well localised and generally not extreme, the results generated by this test sequence are very similar to the Akiyo sequence.

7.2.4 Foreman

7.2.4.1 Characteristics

The Foreman test sequence is a camcorder type output. There is very large local and global motion with a complete scene change. An example image is shown in . Figure 7-8. This sequence is considered one of the most complex and challenging for compression.

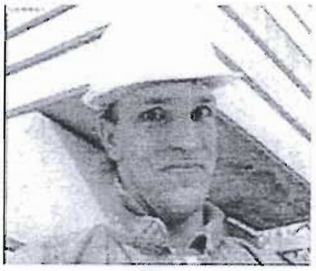


Figure 7-8: Original Foreman Frame #100

7.2.4.2 Computational Complexity

	H.263+		MPEG-4		Proposed
Rate(kbps)	Time(ms)	ME/MC(%)	Time (ms)	ME/MC (%)	Time(ms)
20	4760	67	X	X	2648
25	3156	74	X	X	2800
40	10336	64	X	X	2590
50	15718	70	X	X	4278
284	X	X	44520	85	X

Table 7.8: Complexity Results: Foreman

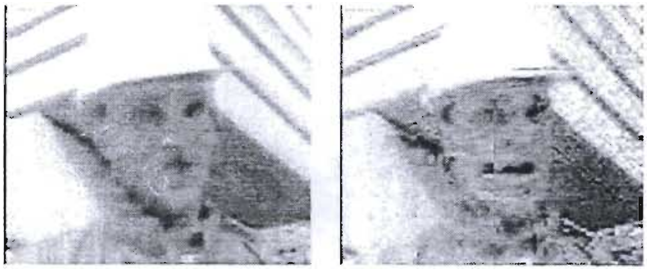
In this case MPEG-4 outputs a 284 kbps stream, the lack of user rate control has prevented this from being altered. This is probably due to distortion control in the MPEG-4 algorithm.

7.2.4.3 RD Performance

The RD performance is given below, MPEG-4 is ommited due to its high rate.

	H.263+		Proposed	
Rate(kbps)	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps
20	27.72	1.1	19.45	10
25	27.45	1.3	20.77	10
40	27.08	2.5	23.42	10
50	27.05	3.4	24.50	10

Table 7.9: RD Performance: Foreman



H.263 Proposed

Figure 7-9: Example Output: Foreman, Frame 100, 40 kbps

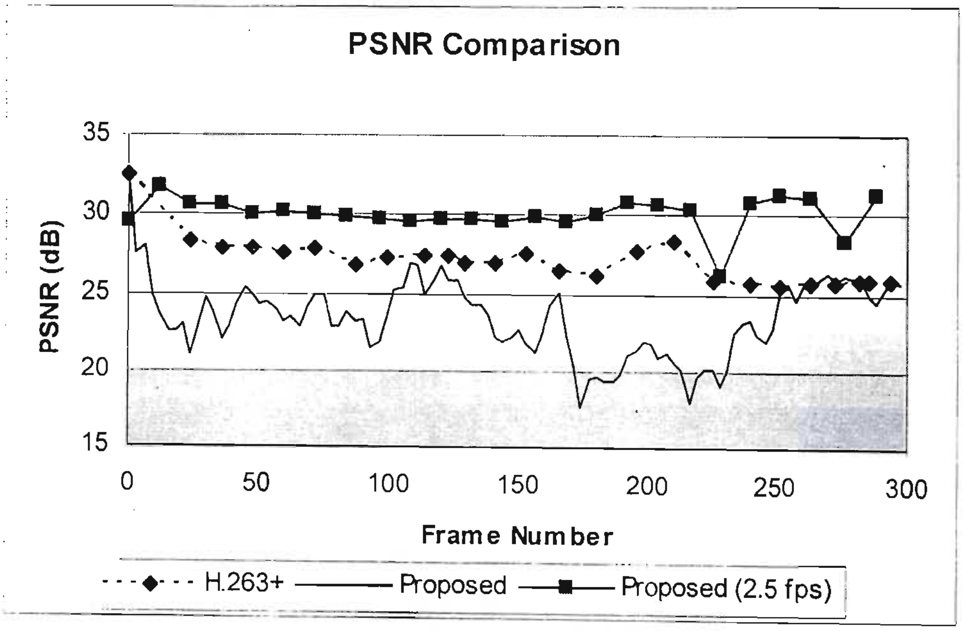


Figure 7-10: Frame by Frame PSNR: Foreman (40 kbps)

7.2.4.4 Discussion

Clearly the proposed algorithm performs very poorly with this test sequence. This is due to one of the assumptions upon which it is built; limited local motion, not holding. There is global motion due to the camera motion and the panning movement during the scene change (starting at about frame #175). This global motion introduces significant coefficients in all the difference image tiles, thus the optimal bit allocation yields no significant RD improvement. In addition the foreman's head occupies much of the frame, and moves considerably, thus distributing significant coefficients across the frame, even though it is a local motion.

MPEG-4 handles this scene by increasing the bit rate to 284 kbps. Again, the lack of user control of this reference software has frustrated efforts to compare it rigorously.

H.263+ in contrast, drops the frame rate, thus gaining both computational and RD performance gain over the proposed algorithm. Although the proposed algorithm cannot automatically adjust the frame rate, it is a simple matter to achieve through user intervention.

The frame rate was adjusted 2.5 fps to match H.263 at 40 kbps. Figure 7-10 demonstrates that this action regains the RD advantage of the proposed scheme. In addition the execution time of the algorithm under this configuration is 863ms, compared to 10336ms for H.263+ for this configuration.

This test sequence demonstrates that in addition to interframe rate control, the proposed algorithm will benefit from frame rate control.

7.2.5 News

7.2.5.1 Characteristics

The News test sequence is a new broadcasting scene. The Akiyo sequence is a detail from this scene. There is no global motion and low local motion. However, the background features a ballet scene, with two dancers who appears and move very vigorously. The sequence was constructed by the MPEG-4 group to demonstrate separate video object encoding. The ballet is the complex visual feature, but is limited to a screen in the background, thus by including it on a separate video plane, coding advantages can be gained. There are four moving objects in this scene, distributed across the frame. An example image is shown in Figure 7-11.

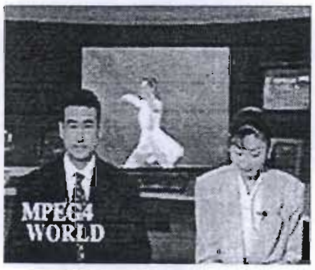


Figure 7-11: Original News Frame #100

7.2.5.2 Computational Complexity

	H.263+		MPEG-4		Proposed
Rate(kbps)	Time(ms)	ME/MC(%)	Time(ms)	ME/MC(%)	Time(ms)
20	39095	87.32	X	X	5626
25	42608	82.77	X	X	5881
40	41066	85.64	59256	88.69	6363
50	44429	79.50	X	X	6927

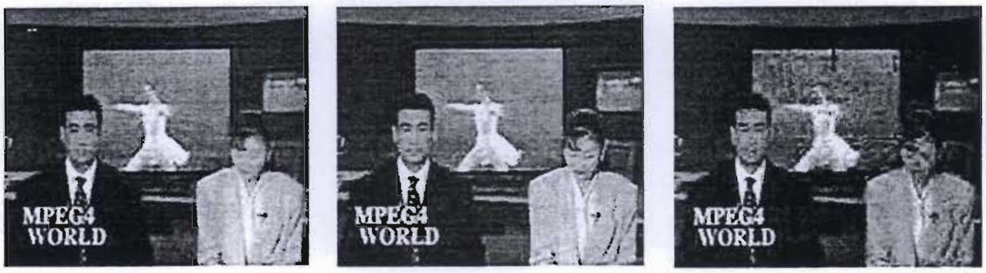
Table 7.10: Complexity Results: News

7.2.5.3 RD Performance

The RD performance is given below.

	H.263+		MPEG-4		Proposed	
Rate(kbps)	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps
20	28.98	10.7	X	X	23.88	10
25	29.97	11.2	X	X	25.36	10
40	31.56	11.6	32.63	10.1	27.81	10
50	32.45	11.7	X	X	29.40	10

Table 7.11: RD Performance: News



H.263 MPEG-4 Proposed

Figure 7-12: Example Output Images: News, Frame 100, 40 kbps

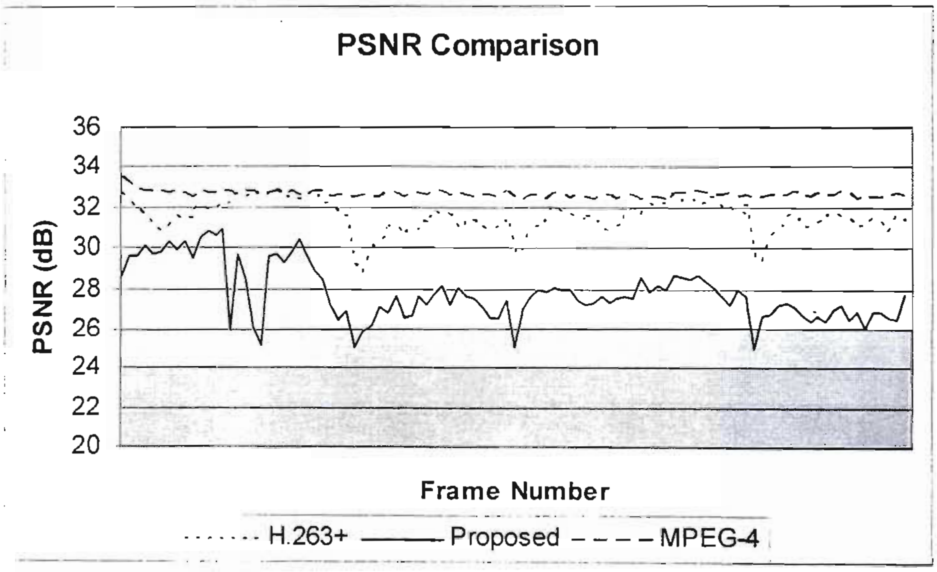


Figure 7-13: Frame by Frame PSNR: News (40 kbps)

7.2.5.4 Discussion

The proposed algorithm fails to encode this sequence efficiently. This is due to the distributed nature of the motion in the scene. Both news readers and the ballerinas are moving simultaneously, and taken together, this results in most of the tiles in the image containing significant coefficients. Thus, although the assumption of coefficient clustering still holds, there are many coefficients clusters and the algorithm is unable to concentrate bits sufficiently to gain advantage over the standards.

The underlying flaw is the spatial syntax. The 16 tile segmentation used here fails to localize the motion sufficiently for there to exist sufficient energy differences between spatial regions for bit allocation to be useful. This motivates further work to develop a new or adaptive syntax that will overcome this shortcoming.

7.2.6 Coastguard

7.2.6.1 Characteristics

The Coastguard test sequence is an outdoor moving scene. The sequence is dominated by a global translation motion. The camera is mounted on one boat, and the scene consists mainly of another boat moving at the same speed in parallel with the camera. Thus the central object does not change much, but the background pans past.

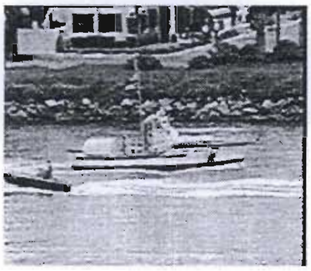


Figure 7-14: Original News Frame #100

7.2.6.2 Computational Complexity

	H.263+		MPEG-4		Proposed
Rate(kbps)	Time(ms)	ME/MC(%)	Time(ms)	ME/MC(%)	Time(ms)
20	2919	41.49	X	X	5762
25	3817	46.36	X	X	5296
40	5351	47.34	X	X	6600
50	4372	47.67	X	X	6969
240	X	X	54241	81.03	X

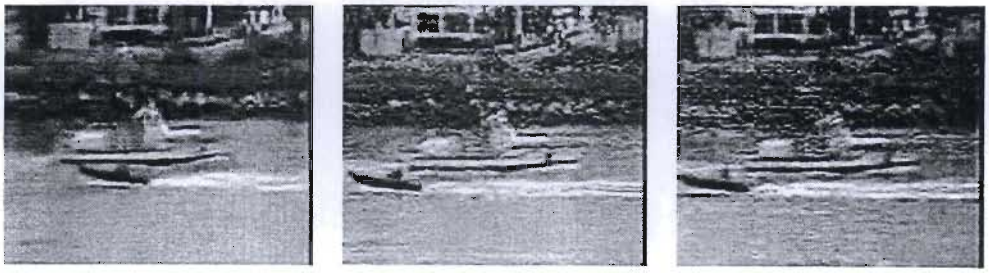
Table 7.12: Complexity Results: News

7.2.6.3 RD Performance

The RD performance is given below.

	H.263+		MPEG-4		Proposed	
Rate(kbps)	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps	Avg. PSNR (dB)	fps
20	25.84	1.2	X	X	20.45	10
25	25.68	1.2	X	X	21.18	10
40	25.45	2.6	X	X	22.90	10
50	25.42	3.7	X	X	23.45	10
240	X	X	30.70	10	X	X

Table 7.13: RD Performance: Coastguard



H.263 Proposed (2.5 fps) Proposed

Figure 7-15: Example Output: Coastguard, Frame 100, 40 kbps

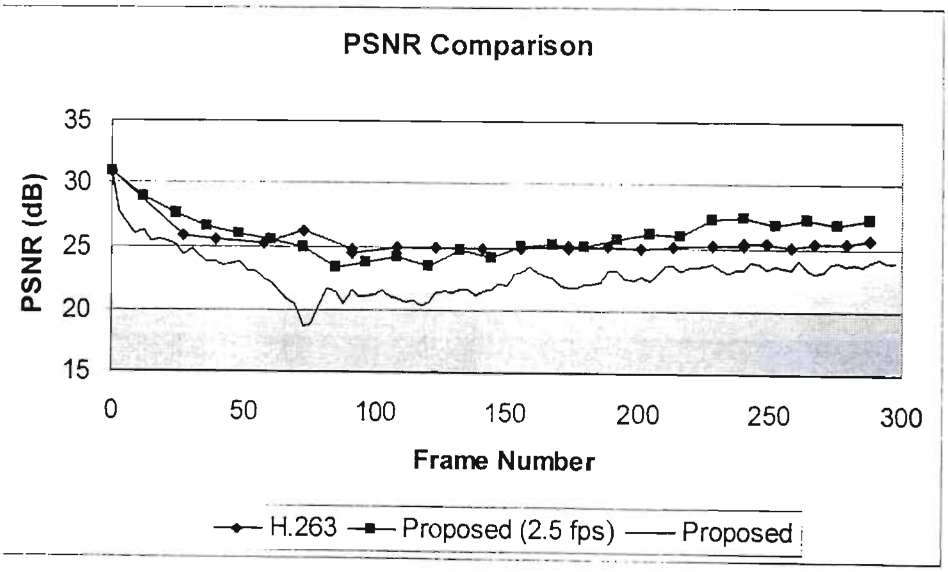


Figure 7-16: Frame by Frame PSNR: Coastguard (40 kbps)

7.2.6.4 Discussion

Once again the global motion conflicts with the assumption of coefficients clustering. As the camera is panning, each tile contains significant motion, and hence significant coefficients. Thus this sequence represents a large departure from the assumptions of the algorithm.

The proposed algorithm presents very poor results. Once again, reducing the frame rate manually to allow a greater number of bits per frame improves the RD performance. The modified algorithm produces a 2.5 fps sequence in 1284ms, with an average PSNR of 26.00dB; outperforming H.263 on both metrics.

Unfortunately, rigorous comparison to MPEG-4 cannot be made for this sequence.

Figure 7-15 for the H.263+ does not present frame #100 as stated. Due to the low frame rate, frame #100 is not actually coded, but the closest frame to #100 is displayed.

7.2.7 Discussion of Comparison to Standards

This discussion is deferred to Section 7.5.1 where an integrate discussion of all the tests is made.

7.3 Comparison to Wavelet Literature

Much of the RD advantage of our scheme over the video compression standards, is due to use of the wavelet transform. Thus it is valuable to compare this method with the wavelet methods proposed in the literature. Unfortunately there are two severe problems in performing this experiment. The source code for the literature methods is unavailable and computational complexity results are generally not published. Thus, comparing the computational complexity can only be done on a basis of informed assumption.

Notwithstanding the above, we shall attempt to compare both the RD and computational performance of our scheme with those from the literature. We shall in each case perform, as closely as possible, the same tests as used to generate the published results, and compare the results. The discussion will proceed, by paper, in the same order as the papers are presented in Chapter 4.

7.3.1 Shen and Delp [Shen99]

This algorithm is fully described in Section 4.3.3.4.

Shen and Delp present results for colour sequences. We have not considered such coding, and thus a comparison cannot be fairly made.

7.3.2 Lin and Gray [Lin01]

7.3.2.1 Computational Complexity

This algorithm is extremely similar to ours. Indeed the only significant differences are that Lin and Gray utilise the H.263 ME/MC engine, as well as both intra and inter frame RD estimation.

Due to the ME/MC, we believe this scheme will exhibit complexity of the order of the H.263 algorithm, hence, significantly more complex than our algorithm.

7.3.2.2 RD Performance

Lin and Gray present coding results, for the first 50 frames of the Carphone and Foreman sequences, both at 10 fps.

	[Lin01]		Proposed	
	PSNR (dB)		PSNR (dB)	
Rate(kbps)	Carphone	Foreman	Carphone	Foreman
20	31	28	24.6	20.8
40	33.5	31	29.2	24.3
50	35	31.5	30.4	25.5

Table 7.14: RD Performance: Lin and Gray

7.3.2.3 Discussion

Although it cannot be proven at this point, it is strongly expected that as [Lin01] utilised the H.263+ ME/MC algorithm, the proposed algorithm will present far more modest computational demands.

As expected, the proposed algorithm holds no RD advantage over [Lin01]. The proposed algorithm is able to present superior RD characteristics to the video standards, due to its use of the wavelet transform, as opposed to the DCT. [Lin01], however also includes the wavelet transform, as well as more advanced RD estimation. Thus it is wholly expected to present superior RD performance.

7.3.3 Marpe and Cycon [Marp99]

7.3.3.1 Computational Complexity

Again, no computational data is published for this algorithm. As with [Lin01] the algorithm performs ME/MC using a block matching scheme very similar to H.263+. Thus we reasonably expect to have computational superiority over this algorithm.

7.3.3.2 RD Performance

Results for this algorithm are published for the Akiyo, Hallmonitor, News and Foreman sequences. Each one is in QCIF format, at various frame rates.

			[Marp99]	Proposed
Sequence	Bit Rate(kbps)	fps	PSNR (dB)	PSNR (dB)
Akiyo	25	10	37.4	32.6
Hallmonitor	25	10	38.9	29.9
News	50	7.5	34.7	30.1
Foreman	50	7.5	32.0	25.4

Table 7.15: RD Performance: Marpe and Cycon

7.3.3.3 Discussion

Once again the proposed algorithm is outperformed by the literature method, and this is probably due to the inclusion of ME/MC in the literature method.

7.3.4 Yang and Ramchandran [Yang00]

7.3.4.1 Computational Complexity

Computational complexity is difficult to ascertain for this method, as it is based on hierarchical motion estimation. Although this entails multiple ME/MC steps, the authors claim that the computational complexity of their backward estimation is equal to that of a block matching search. In addition their forward estimation incurs an additional 20%-30% penalty. Based on this, we may estimate that the complexity of this algorithm is similar or, probably, greater than that of the H.263+ and MPEG type schemes. Therefore our algorithm should be substantially faster.

7.3.4.2 RD Performance

Results for this algorithm are published for the Mother and Daughter, Miss America and Football test sequence, however only the Mother and Daughter sequence could be obtained for comparison.

			[Yang00]	Proposed
Sequence	Bit Rate(kbps)	fps	PSNR (dB)	PSNR (dB)
M & D	48	15	35.6	33.0

Table 7.16: RD Performance: Yang and Ramchandran

7.3.4.3 Discussion

As in the previous cases this method outperforms the proposed algorithm in an RD sense, at the cost of computational overhead. This is despite the test sequence adhering to the strictly local motion assumptions of our algorithm. As both algorithms use wavelet coding, the utility of the ME/MC for temporal coding can be clearly seen.

7.3.5 Kim, Xiong and Pearlman [Kim00]

7.3.5.1 Computational Complexity

This paper discusses the computational complexity of the algorithm. When a pure 3D transform is applied without ME/MC the algorithm performs 2.53 times faster than H.263 (for the Carphone sequence at 10 fps, 30 kbps). This performance gain is less than with our algorithm, which usually operates between 6 and 10 times faster than H.263.

In addition the 3D SPIHT algorithm requires the buffering of a 16 frame group of frames. This memory overhead is usually unacceptable for embedded devices. Our algorithm requires only buffering of the current and previous frames.

7.3.5.2 RD Performance

There are several results published for this algorithm.

			[Kim00]	[Kim00] (ME/MC)	Proposed
	Bit Rate(kbps)	fps	PSNR (dB)	PSNR(db)	PSNR (dB)
Carphone	30	10	30.22	30.39	26.03
Carphone	60	10	32.97	33.19	30.08
M & D	30	10	31.71	32.78	31.86
M & D	60	10	35.57	35.69	35.11
Hallmonitor	30	10	32.95	32.30	30.89
Hallmonitor	60	10	37.36	37.95	34.72

Table 7.17: RD Performance: Kim et al

[Xu01] presents further results for this algorithm, considering only the first 96 frames of the following sequences.

			[Kim00]	Proposed
	Bit Rate(kbps)	fps	PSNR (dB)	PSNR (dB)
Akiyo	20	30	29.43	X
Akiyo	40	30	33.38	28.83
Akiyo	40	10	X	36.18
M & D	20	30	30.09	X
M & D	40	30	33.33	27.75
M&D	40	10	X	33.40
Coastguard	40	30	25.88	19.42
Coastguard	80	30	27.65	23.32
Coastguard	40	10	X	22.90

Table 7.18: RD Performance: Kim et al, further results

7.3.5.3 Discussion

This algorithm displays the usual pattern of outperforming the proposed scheme on RD terms, but not on computational terms. In addition to the usual computational complexity, the 3D SPIHT algorithm presents severe memory demands.

The proposed algorithm is unable to encode at 20 kbps and 30 fps, as the algorithm requires a minimum of ρ=0.99 per tile, which sets a minimum possible number of bits per frame. The set of results presented in [Xu01] is rather unrealistic, as the requirement for low-bit, high-frame rate video is hard to justify.

I believe that 3D transform based schemes probably require high inter-frame correlation in order to present competitive results, due to the application of the wavelet transform in the temporal direction. Thus results might be published for 30 fps sequences, as this is where the highest performance is found.

7.3.6 Xu, Xiong, Li and Zhang [Xu02]

7.3.6.1 Computational Complexity

The expected computational complexity of this scheme is large, compared to our algorithm, due to the 3D wavelet transform, as is the case for the previous algorithm, [Kim00].

The memory burden of a 16 frame frame-buffer is also excessive, for our application.

7.3.6.2 RD Performance

Once more the published results are for a 30 fps sequence.

			[Kim00]	Proposed
	Bit Rate(kbps)	fps	PSNR (dB)	PSNR (dB)
Akiyo	42	30	35.02	27.22
Akiyo	40	10	X	36.18
Coastguard	45	30	27.10	X
Coastguard	40	10	X	22.90

Table 7.19: RD Performance: Kim et al

7.3.6.3 Discussion

Our algorithm fails to compete for the Coastguard sequence, as previously mentioned; the global motion of this sequence breaks the assumptions of our algorithm.

For Akiyo the proposed algorithm is unable to compete at 30 fps, but demonstrates a recovery for 10 fps, as discussed for [Kim00]. Obtaining RD results for the 3D transform schemes at 10 fps would be useful for comparison at lower interframe correlation.

7.3.7 Wang, Xiong, Chou and Mehrotra [Wang02]

This algorithm and the proposed algorithm cannot be reasonably compared. [Wang02] presents an algorithm designed expressly for scenes dominated by global motion, and presents results only for such scenes, whereas the proposed algorithm is designed for global motion free scenes. Thus no comparison is offered.

7.4 Component Analysis

This section will review the components of the system and discuss the contribution each makes to the RD performance and computational demand of the proposed system. This is undertaken in order to discover areas of improvement for future work. The Microsoft Profiler is used to measure the amount of time spent in each algorithm in the scheme, the results of this measurement are given for each of the tested sequences. This is a slightly problematic approach to measurement as the Windows XP platform is multitasking, thus background processing may interfere with the measurements.

Section 7.4.1 presents the component complexity results. Section 7.4.2 presents a discussion of the contribution of each of these components to the overall RD performance. The discussion is deferred to Section 7.5.3.

7.4.1 Component Timing Data

The following tables indicate the amount of time spent in each algorithm of the scheme, during coding of various test sequences at several output bit rates.

	20 kbps		25 kbps		40 kbps		50 kbps	
	Time (ms)	%	Time (ms)	%	Time (ms)	%	Time (ms)	%
Total	2889.662	100	2889.662	100	2967.622	100	6209.401	100
RD Estimation	337.262	11.67133	337.262	11.67133	342.957	11.55663	345.534	5.564691
Wavelet	278.297	9.63078	278.297	9.63078	278.752	9.39311	295.807	4.763857
SPIHT	1223.199	42.33018	1223.199	42.33018	1268.274	42.73705	4201.994	67.67149
Arith Coder	67.922	2.350517	67.922	2.350517	67.503	2.27465	329.684	5.309433
Sundry		34.0172		34.0172		34.03857		16.69053

Table 7.20: Component Timing: Akiyo

	20 kbps		25 kbps		40 kbps		50 kbps	
	Time (ms)	%	Time (ms)	%	Time (ms)	%	Time (ms)	%
Total	2725.425	100	2865.381	100	3344.42	100	6329.125	100
RD Estimation	333.055	12.2203	332.803	11.61462	344.029	10.28666	342.613	5.413276
Wavelet	276.227	10.13519	289.462	10.10204	284.659	8.511461	286.694	4.529757
SPIHT	1062.648	38.99018	1163.038	40.5893	1283.505	38.37751	4258.912	67.29069
Arith Coder	37.787	1.386463	54.62	1.906204	64.215	1.920064	145.663	2.301471
Sundry		37.26788		35.78784		40.90431		20.4648

Table 7.21: Component Timing: Hallmonitor

	20 kbps		25 kbps		40 kbps		50 kbps	
	Time (ms)	%	Time (ms)	%	Time (ms)	%	Time (ms)	%
Total	2648.391	100	2800.165	100	2590.423	100	4278.459	100
RD Estimation	332.489	12.55438	331.807	11.84955	325.84	12.57864	327.778	7.661123

Wavelet	273.418	10.32393	277.745	9.91888	275.707	10.64332	285.245	6.667003
SPIHT	982.709	37.10589	1068.831	38.17029	924.324	35.68236	2465.403	57.62362
Arith Coder	28.926	1.09221	81.995	2.92822	53.178	2.052869	164.704	3.84961
Sundry		38.9236		37.13306		39.04281		24.19864

Table 7.22: Component Timing: Foreman

	20 kbps		25 kbps		40 kbps		50 kbps	
	Time (ms)	%	Time (ms)	%	Time (ms)	%	Time (ms)	%
Total	5626.883	100	5881.206	100	6363.696	100	6927.499	100
RD Estimation	339.954	6.041604	336.748	5.725832	345.436	5.428229	340.782	4.919265
Wavelet	283.181	5.032644	279.383	4.750437	285.541	4.487031	287.326	4.147615
SPIHT	3639.914	64.68793	3807.643	64.74255	4196.695	65.94745	4666.601	67.36343
Arith Coder	143.915	2.557633	180.397	3.067347	248.554	3.905812	312.19	4.506533
Sundry		21.68019		21.71383		20.23148		19.06316

Table 7.23: Component Timing: News

	20 kbps		25 kbps		40 kbps		50 kbps	
	Time (ms)	%	Time (ms)	%	Time (ms)	%	Time (ms)	%
Total	5762.062	100	5925.95	100	6599.903	100	6968.925	100
RD Estimation	336.556	5.840895	331.147	5.588083	332.463	5.037392	337.065	4.836686
Wavelet	287.123	4.98299	283.194	4.778879	287.28	4.352791	285.639	4.098753
SPIHT	3736.256	64.84234	3872.501	65.34819	4439.628	67.26808	4770.421	68.45275
Arith Coder	133.771	2.321582	176.106	2.971777	254.607	3.857739	290.691	4.171246
Sundry		22.01219		21.31307		19.484		18.44056

Table 7.24: Component Timing: Coastguard

7.4.1.1 Component Complexity Discussion

7.4.1.1.1 Wavelet Transform

The wavelet transform exhibits an almost constant execution time of 300ms. This is logical as the wavelet transform stage is identical for all bit rates and input data. Slight variations that occur are probably the result of the imperfect measurement method.

7.4.1.1.2 RD Estimator

As with the wavelet transform, the execution time of the RD estimator is independent of the output bit rate. As this stage occurs before any quantization, or data reduction has occurred, this is natural.

7.4.1.1.3 SPIHT

These tables show the overriding contribution of the SPIHT algorithm to the system complexity. The SPIHT and arithmetic coder execution times are seen to increase with bitrate, as more data is processed.

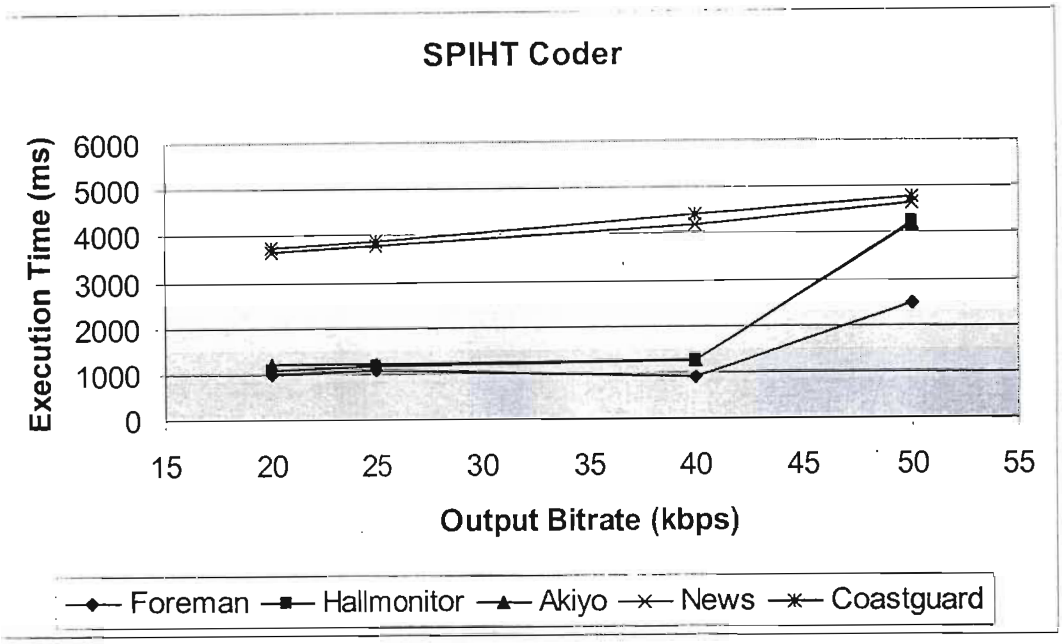


Figure 7-17: SPIHT Execution Time vs Output Bit Rate

Figure 7-17 above is most enlightening. For the first three scenes, Foreman, Hallmonitor and Akiyo, the execution time is fairly constant, until 40 kbps, at which point the execution time increases dramatically. News and Coastguard present smoother increasing functions. This behaviour may be explained through the action of the Optimal Bit Allocation unit.

As mentioned in Section 7.2, News and Coastguard produce significant coefficients in many of the tiles, due to multiple local motions, and global motion respectively. Thus the bit allocation unit spreads the bits fairly evenly between the tiles. For each coefficient encountered the SPIHT algorithm must perform a full tree search to decide whether to descend the tree. or zerotree quantize.

Hallmonitor and Akiyo present very localized motion, thus the bit allocation unit causes very few tiles to consume most of the bit budget. For the bit starved tiles, very few coefficients will be encountered by the SPIHT algorithm before the bit budget is expended, thus very few tree searches will occur for these tiles. For those tiles allocated many bits, the SPIHT algorithm will have the opportunity to descend the wavelet trees. At each level of descent the full tree search becomes faster, as the remaining wavelet tree is smaller than the original.

Thus for the Hallmonitor and Akiyo sequences, the low bit rate will cause the optimal bit allocation unit to allocate most bits to very few tiles, owing to the strong coefficient clustering in these sequences. Thus the SPIHT process will be characterized by few tree searches in total, and 'deep' (as far as allowed by the bit rate) tree descents for those coefficients considered. As mentioned, this is an efficient manner for the SPIHT algorithm to operate, as the tree searching represents the main time consumption of the algorithm. For global motion scenes, however, the bit budget is evenly spread over the tiles, and thus there will be many tree searches, which is inefficient. This behaviour indicates why Hallmonitor and Akiyo are significantly faster than News and Coastguard at low bit rates.

For Hallmonitor and Akiyo, as the bit rate increases, it becomes RD optimal to include more tiles in the output stream. This causes an increase in the number of tree searches, and hence the complexity. This causes the behaviour seen at 40 kbps in Figure 7-17.

The Foreman sequence seems anomalous. Foreman presents large local and global motions, it is postulated that the large motion of the man in the centre of the frame dominates the difference frame, and hence the effect discussed for local motion scenes is exhibited. However, the sequence is very complex and this explanation merely suggestive.

7.4.1.1.4 Arithmetic Coder

The arithmetic coder execution times increase similarly to the SPIHT times.

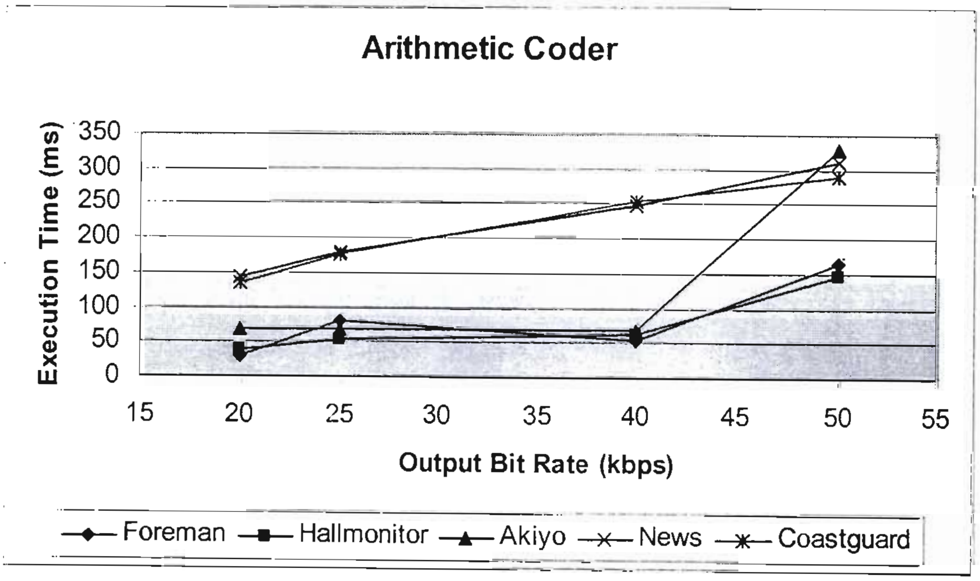


Figure 7-18: Arithmetic Coder Execution Time vs Output Bit Rate

The arithmetic coder is clearly reacting to the output stream of the SPIHT encoder stage. Based on the previous discussion it is hypothesized that at higher rates, more tiles produce sufficiently long SPIHT output streams to be deemed complex; the arithmetic coder is thus responding to the complexity of this stream. However, as the arithmetic coder is not written by this author, an precise explanation is not offered.

7.4.1.1.5 Component Complexity Summary

The previous sections show that the SPIHT algorithm is the most complex and unpredictable of the system components. For the test sequences above, SPIHT is responsible for 56% of the computation time, on average. This motivates for future work to decrease this complexity, or for abandonment of the SPIHT algorithm in favour of an ECECOW-like (Section 3.3.9) system. This will be explored in the final discussion.

7.4.2 Component RD Contribution

This section will explore the RD effect of various design decisions taken. Specifically the use of difference frames will be explored in Section 7.4.2.1. Sections 7.4.2.2 and 7.4.2.3 will investigate the benefits of optimal bit allocation and arithmetic coding respectively. Finally Section 7.4.2.4 will discuss the effect of tiling.

Two sequences will be used in this discussion; Hallmonitor, as surveillance is our intended application and the sequence fulfils all the assumptions made in the algorithmic design, and Coastguard as it fulfils none of the assumptions. The different behaviour of these two sequences can then be used to verify our assumptions. All tests were done at 10 fps.

7.4.2.1 Difference Frames

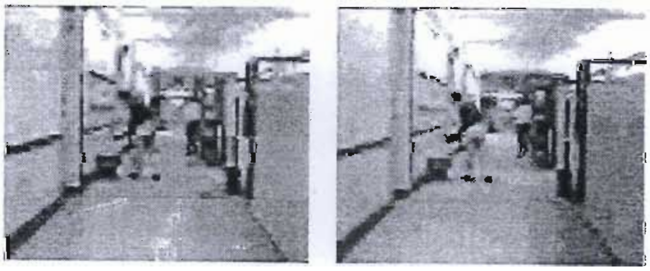
In order to test the utility of difference frames, the proposed algorithm is converted to an intra-frame coding algorithm, by removing the frame differencing. Each frame in the sequence is simply coded with the SPIHT / arithmetic coder pair. In addition, no optimal bit allocation is performed. This algorithm thus represents the baseline of performance.

For the Hallmonitor sequence the following results are produced:

	PSNR (dB)				
Rate (kbps)	20	25	50	100	200
Proposed	28.88	29.90	35.11	X	X
Intra – 16 Tile	X	8.52	16.82	22.22	27.48
Intra – 1 Tile	X	10.47	17.99	25.19	29.99

Table 7.25: Intra-frame RD Results - Hallmonitor

For visual reference, the intraframe output at 100 kbps, is shown in Figure 7-19.



16 Tiles 1 Tile

Figure 7-19: Intra-frame Visual Results – Hallmonitor (Frame #100, 100 kbps)

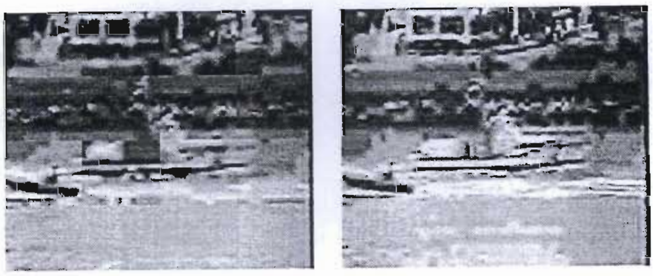
Similarly, for Coastguard:

	PSNR (dB)				
Rate (kbps)	20	25	50	100	200
Proposed	20.45	21.18	23.45	X	X

Intra – 16 Tile	X	9.99	18.39	22.67	27.01
Intra – 1 Tile	X	14.87	21.12	24.83	28.88

Table 7.26: Intra-frame RD Results - Coastguard

For visual reference, the intraframe output at 100 kbps, is shown in Figure 7-20.



16 Tiles 1 Tile

Figure 7-20: Intra-frame Visual Results – Coastguard (Frame #100, 100 kbps)

7.4.2.1.1 Difference Frame Discussion

An immediate observation is that the gain from using difference frames is enormous. For the Hallmonitor sequence, virtually the same distortion is achieved by the proposed algorithm at 25 kbps as the intra-frame coder achieves at 200 kbps.

Furthermore the increase in RD performance is significantly greater for the Hallmonitor sequence, than for the Coastguard sequence. At 50 kbps the Hallmonitor gains 17.22 dB PSNR (virtually doubling) with the proposed scheme, whereas the Coastguard sequence gains only 2.33dB. This result confirms the hypothesis that local motion leads to strong coefficient clustering, which may be efficiently captured through tiling and optimal bit allocated coding, the fundamental assumption of the scheme.

A final observation is that for the intra-frame coder, tiling has the effect of decreasing the PSNR. This is due to the basic notion that the smaller the data source, the less the ratio to which it may be compressed. Incidentally this is the fundamental limitation of the original JPEG algorithm, which uses 8x8 pixel tiles, and at high compression ratios simply runs out of coefficients to quantise in each block. This effect will be further explored in Section 7.4.2.4.

7.4.2.2 Optimal Bit Allocation

In order to test the concept of optimal bit allocation, this unit is removed from the system. The system thus becomes a difference frame coder, followed by SPIHT and arithmetic coding. The RD performance is shown below.

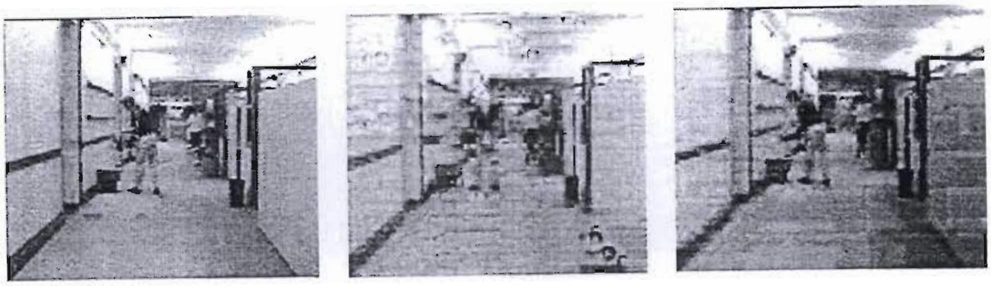
The performance for Hallmonitor is shown below:

	PSNR (dB)				
Rate (kbps)	20	25	50	100	200
Proposed	28.86	29.89	35.11	X	X

16 Tile	X	21.75	28.77	33.96	40.45
1 Tile	X	24.12	32.18	40.44	45.94

Table 7.27: Unoptimised RD Results - Hallmonitor

For visual reference, the output at 25 kbps, is shown in Figure 7-21.



Proposed 16 Tiles 1 Tile

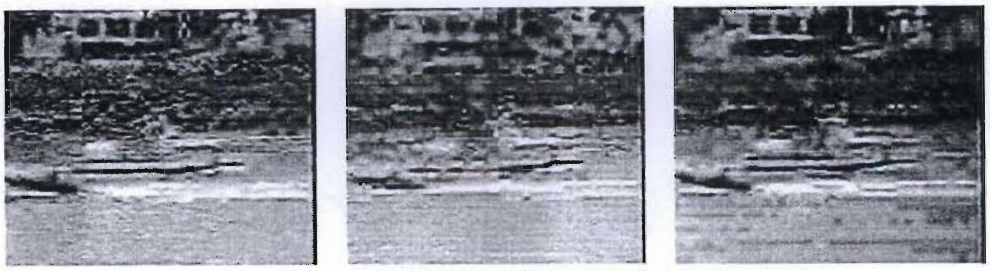
Figure 7-21: Unoptimised Visual Results – Hallmonitor (Frame #100, 25 kbps)

Similarly, for Coastguard:

	PSNR (dB)				
Rate (kbps)	20	25	50	100	200
Proposed	20.45	21.18	23.45	X	X
16 Tile	X	20.10	22.85	25.18	28.54
1 Tile	X	21.62	23.84	26.13	29.18

Table 7.28: Unoptimised RD Results – Coastguard

For visual reference, the output at 25 kbps, is shown in Figure 7-22.



Proposed 16 Tiles 1 Tile

Figure 7-22: Unoptimised Visual Results – Coastguard (Frame #100, 25 kbps)

7.4.2.2.1 Optimal Bit Allocation Discussion

This test highlights the trade-off between tiling and optimal bit allocation. For the Coastguard sequence, the straight single tile difference frame coder outperforms the proposed system. This is owing to the effect of tiling reducing the information available to the SPIHT and arithmetic coders, thus reducing the RD performance. This result may be explained by the fact that optimal bit allocation is ineffective for the Coastguard sequence owing to the global motion, as previously discussed.

The Hallmonitor sequence demonstrates the positive side of the tradeoff. Here the optimal bit allocation is expected to be successful, and the proposed system outperforms the difference frame coder.

These results serve to confirm that difference frame coding, coupled with optimal bit allocation between tiles, is effective in describing local motion scenes.

7.4.2.3 Arithmetic Coding

The final stage is the arithmetic coder. The RD contribution of this algorithm, is tested by outputting the total bit rate, before and after the arithmetic coder.

The following two tables, show for a particular value of distortion, the total bit rate, before and after arithmetic coding.

	Rate (kbps)			
PSNR (dB)	28.9	29.9	33.8	35.1
Proposed	20.9	24.3	39.14	50.3
w/o Arith. Coder	40.8	46.4	66.3	80.2

Table 7.29: Arithmetic Coder RD Contribution: Hallmonitor

	Rate (kbps)			
PSNR (dB)	20.45	21.2	22.9	23.5
Proposed	19.8	24.9	40.4	49.4
w/o Arith. Coder	36.2	44.1	65.7	77.4

Table 7.30: Arthimetic Coder RD Contribution: Coastguard

7.4.2.3.1 Arithmetic Coder Discussion

The two tables above show that the arithmetic coder reduces the output data stream by 40 to 50%, without affecting the PSNR.

7.4.2.4 Tiling

It was shown in Section 7.4.2.2 that tiling reduces the RD performance of the system, but in the case of local motion provides the spatial syntax to allow optimal bit allocation. Thus there will exist a trade-off, for local motion scenes, between the fine spatial resolution provided by small tiles, and the limited compression such tiles impose.

Although this effect was noted, modifying the algorithm to allow optimal bit allocation for different tile sizes requires significant effort in reconfiguring the RD estimator and modifying the control code. Thus this work was not undertaken.

Furthermore tile-size also effects the number of decomposition levels possible in the wavelet transform stage. As the tile size decreases, the number of coefficients in a single row or column of the lowest resolutions subband approach the length of the wavelet transform filter. This effect naturally reduces the efficacy of the process, and if the number of coefficients becomes less than the filter length, renders the filter useless.

It is noted that future work centred around studying the effect of tile size on performance for different sequence types, and hence an adaptive tile configuration, may yield coding benefit.

7.5 Discussion

This section presents a unified discussion of the all the tests undertaken in this chapter.

7.5.1 Comparison to Standards

The proposed algorithm compares admirably to the video compression standards. In cases where there is no global motion, and limited local motion, such as Hallmonitor and Akiyo, the algorithm outperforms the standards. In the cases of the News sequence, there are multiple local motion sources, thus the significant coefficients are spread over the frame and the tiling spatial syntax is unable to localize the motion. Thus the algorithm fails to code this sequence efficiently.

In the case of global motion, the proposed algorithm presents poor RD performance. However, by implementing a frame rate control (manually at this point) the algorithm may be made competitive with H.263. However, it is believed this is simply because H.263 handles global motion poorly. Systems such as [Wang02] with proper global ME/MC are expected outperform our scheme significantly.

This algorithm implements no inter-frame rate control, which could allocate rate between frames in order to maintain a constant PSNR. The algorithm has been designed instead to output a constant bitrate, in order to maintain a constant frame rate over a fixed bandwidth radio channel. Rate control requires a deep buffer at the receiver in order to smooth the frame rate; this is in opposition to the specification of our project. However, as RD estimates of each frame exist, implementing inter-frame rate control should be a trivial matter should this specification change.

In addition to bit rate control, the Foreman sequence demonstrates that the algorithm will benefit from frame rate control. Again, the availability of an accurate RD estimate trivializes the automation of this functionality.

The overriding observation is that the proposed scheme has a significantly lower execution time than the standards, as expected from the system design. In addition, it outperforms the standards, on RD criteria for all but one test sequence. It is expected that this RD performance may be on account of the simple H.263+ configuration chosen. Other configurations may present better performance, however, as the literature review of Chapter 4 shows even the highest best reported RD performance is comparable to that of our algorithm. Furthermore, this increases RD performance will be at the expense of increased complexity.

These tests show that for a restricted, but useful, application this algorithm executes approximately ten times faster than the compression standards algorithm, and performs comparably.

7.5.2 Comparison to Wavelet Literature

The proposed algorithm presents poor RD performance compared to the other wavelet techniques reviewed. The algorithm draws RD advantage over DCT based schemes through the use of the wavelet transform. This advantage is naturally lost when competing with other wavelet based systems, which have the further benefit of ME/MC.

However, although unconfirmed by testing, it is strongly expected that the proposed algorithm will present far lower computational demands than the wavelet based schemes, owing to avoiding ME/MC required by most of these schemes and the large memory buffers required by the 3D transform based schemes.

7.5.3 Component Analysis

A component analysis was undertaken to test the assumptions made during the algorithmic design.

Section 7.4.1 finds that SPIHT is the most significant contributor to the system complexity, being responsible for an average 56% of the execution time. The other significant blocks, such as the wavelet transform, RD estimator and arithmetic coder usually account for 25% or less of the execution time, depending on the total time. More meaningfully, the RD estimator and wavelet transform consume between 600 and 650 ms, regardless of the sequence or rate. The arithmetic coder execution time is dependant on the source and rate, but accounts for between only 1 and 5% of the execution time.

The remaining time is spent in I/O and memory manipulations. A recent review of the source code has revealed that significant savings may be made in these areas.

7.5.4 Performance / Complexity Tradeoff

7.5.4.1.1 Contribution of RD Estimation and Optimal Bit Allocation

This section will demonstrate the RD and PSNR effect of each block in the proposed system.

The following figure shows the average PSNR for various block combinations for the Hallmonitor sequence, coded at 50 kbps.

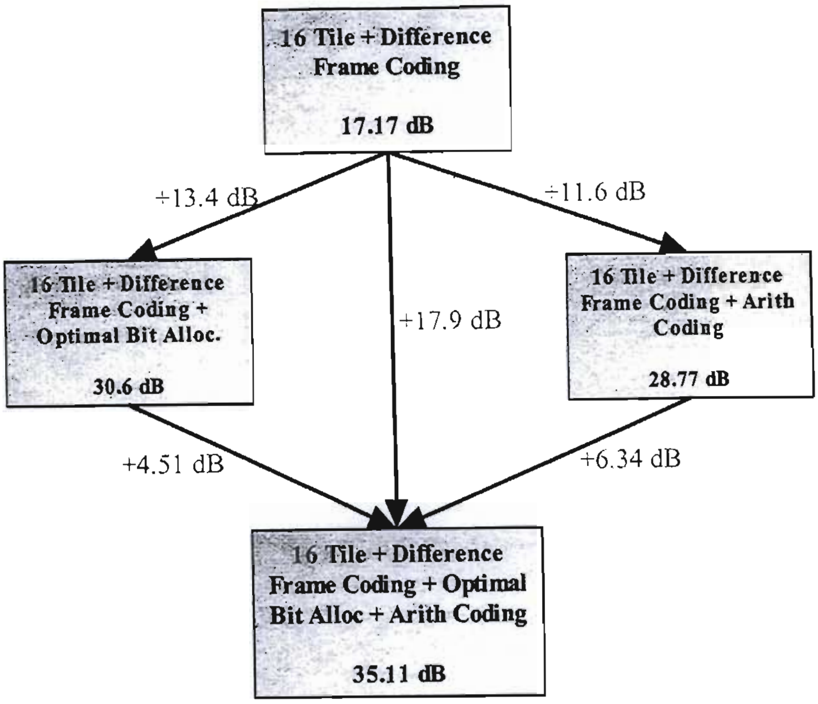

16 Tile + Difference Frame Coding
17.17 dB
+13.4 dB
+11.6 dB
16 Tile + Difference Frame Coding + Optimal Bit Alloc.
30.6 dB
+17.9 dB
16 Tile + Difference Frame Coding + Arith Coding
28.77 dB
+4.51 dB
+6.34 dB
16 Tile + Difference Frame Coding + Optimal Bit Alloc + Arith Coding
35.11 dB

Figure 7-23: Block PSNR Contribution: Hallmonitor

The rightmost block in Figure 7-23 is an algorithm that can be built from standard components. The +6.34 dB PSNR gain to the bottom most configuration, is as a direct result of the work designed and implemented in Chapters 5 and 6.

7.5.4.1.2 Comparison of Temporal Decorrelation Methods

This section will attempt to compare the utility of the ME/MC based temporal coding, the difference coding, and optimal bit allocation strategy.

The following table compares the performance and complexity of an intra-frame coder, the best wavelet video compression system, H.263 and the proposed algorithm for the Hallmonitor sequence at 25 kbps, 10 fps.

	PSNR (dB)	Complexity
Intraframe Coder	10.47 (Sec. 7.4.2.1)	X
Best Literature Method [Marp99]	38.9 (Sec. 7.3.3)	5X
H.263+	30.0 (Sec. 7.2.3.3)	5X
Proposed Method	29.9 (Sec. 7.3.3.3)	1.1X

Table 7.31: Performance / Complexity Trade-off Comparison

The complexity estimate is taken from the following argument. In H.263+, ME/MC accounts for 80% of the execution time, which is taken to be a measure of the complexity. [Marp99] also relies on ME/MC and although it includes several other features, the complexity is conservatively estimated to be equivalent to H.263+. The major difference between the intra-frame coder of Section 7.4.2.1 and H.263+ is the ME/MC, thus it is taken as the major complexity increase. Thus, in [Mar99] and H.263+, it is estimated that ME/MC accounts for

80% of the computational load and the remaining 20% is spent in algorithms common with the intra-frame coder. Thus [Marp99] and H.263+ are 400% more complex than the intra-frame coder. By the same argument, that the RD estimation and optimal bit allocation are the only major additions to intra-frame coding for our scheme, and that these algorithms account for 10% of the total complexity, leads to the conclusion that our scheme is only 11% more complex than the inter-frame coder.

These results, in conjunction with Table 7.31, indicates that temporal decorrelation through ME/MC presents a 28.4dB gain, with a 400% complexity increase, while our proposed method affords a 19.dB gain, with only an 11% performance increase.

This result is for the Hallmonitor sequence, at 25 kbps. However, a similar relationship is expected for all limited local motion dominated scenes, at any bit rate.

Review of the various images printed in this chapter confirms the rule of thumb, that a 30dB image is 'acceptable' quality. Images at 25dB and below, are almost un-viewable. Thus for limited local motion scenes, the image quality is acceptable at 'low bit rate' (25 kbps and below). Thus for these scenes there is a real gain to using the proposed method, even against the best performing algorithms known to the author.

For significantly distributed local motion, or global motion, although the proposed algorithm has a computational advantage, the image quality falls below 30dB, thus the output is of little value as it is overly distorted.

Thus, for local motion sequences the sequence is approximately 10 times faster, while presenting RD performance that is acceptable for the application.

7.5.5 General Discussion and Conclusion

The proposed algorithm performs similarly to the compression standards on all but one of the sequences tested. Despite a lack of ME/MC, the advantage given by the wavelet transform based quantization and optimal bit allocation allows the RD losses incurred by abandoning ME/MC to be recouped.

However the proposed algorithm presents poorer RD results than the other wavelet based video compression systems. This is because in these cases the lack of ME/MC is not compensated by the advantages of the wavelet transform, and the proposed system is simply insufficiently sophisticated to compete.

However, despite the often mediocre RD performance, the proposed algorithm demands significantly less computational resources than any of the other algorithms proposed by the standards bodies, or presented in the literature. Furthermore this performance is acceptable for the specified application.

There exist opportunities to increase the performance of the system by incorporating frame rate control, and inter frame bit rate control. In addition, as SPIHT represents the performance bottleneck of the system, improvements to this algorithm, or its replacement by another wavelet quantization scheme, may yield significant complexity savings.

The proposed algorithm presents acceptable (~30dB and greater) visual quality, at low bit rates (~25 kbps) for local motion scenes. Thus, for these scenes the significant complexity advantages of the proposed scheme may be realized in a practical setting such as surveillance systems, or mobile devices.

This computational advantage is crucial, as all the other schemes examined cannot be implemented on embedded hardware, owing to either computational or memory excess. There exists a strong demand for an algorithm simple enough for embedded implementation, and our algorithm fulfils this requirement.

Thus the proposed algorithm has is competitive in the coding of local motion scenes for complexity limited systems. Possible real world applications are digital security cameras, or on mobile devices such as PDA's or cell phones.

Chapter 8 - Conclusion

This final chapter will review the findings and conclusions of each previous chapter. Thereafter the proposed algorithm will be discussed and future work will be projected. Finally, a brief conclusion will be offered.

8.1 Chapter Summaries

8.1.1 Chapter 2 – RD Estimation and Optimisation

The main contribution of this chapter is an introduction to the ρ-domain RD estimation method, and the Lagrange method of optimal bit allocation.

This chapter builds a context for the ρ-domain method through presenting the classical RD treatment of Shannon, and the methods of RD estimation employed in current standards. The problems of poor source modelling and the large computational complexity of these methods are highlighted. The ρ-domain method is found to be both fast and accurate.

The Lagrange optimization method is presented from first principles and its optimality for the scheme to be presented in later chapters proven. The justification for using this method is based on its implementation ease, as evidenced by widespread use in the methods presented in literature.

This chapter suggests that combining ρ-domain RD estimation with the Lagrange method may offer an opportunity for truly RD optimal coding, within acceptable computational limits. This combination forms the basis of the video compression scheme proposed Chapter 6.

8.1.2 Chapter 3 – Wavelet Still Image Compression

This chapter presents a broad literature review of the methods of image compression using the wavelet transform.

This chapter finds that modifying either the wavelet basis set or decomposition structure is of little benefit for image compression.

A large number of proposed wavelet quantization strategies are explored. In each case the principles and underlying models are highlighted and discussed.

It found that all the gains made from subband-coding, to zero-tree coding, to higher order context modelling, rely on increasingly accurate image models. This observation is the key design principle employed in the design of the proposed algorithm.

8.1.3 Chapter 4 – Wavelet Video Compression

Chapter 4 reviews the state of wavelet video compression, with a thorough review of the principles and algorithms under research.

The operating principles of the current standards; H.263, MPEG-2 and MPEG-4, are all briefly discussed to introduce major concepts as well as set the stage for the wavelet methods. Their performance analysis is deferred to Chapter 7.

Several leading techniques of wavelet based video compression are then explored. The modification of wavelet methods to allow their use of block-based ME/MC, the new methods of hierarchical ME/MC and 3D subband coding are then discussed, with the performance of such algorithms presented. It is found that block based ME/MC systems currently produce the best results, notwithstanding the problems that blocking introduce.

It is found in this chapter that the wavelet methods are competitive with the standards. However, the exercise of ME/MC in this domain is still rudimentary, although new techniques are promising greater performance

8.1.4 Chapter 5 – Premise of Proposed Algorithm

This chapter proposes a new algorithm for wavelet video compression.

A complexity study of the standard methods is undertaken and it is found that block ME/MC accounts for at least 80% of the execution time of these algorithms. As system complexity is the primary concern of this project, this effect is significant. Based on this finding, it was decided to abandon block ME/MC in order to attempt to find a computationally simpler approach. This decision provided the basis for the work to follow.

The only other alternative presented in the literature is 3D coding, which is extremely memory intensive, and was thus rejected on complexity grounds. The only remaining option is difference frame coding, and this method was thus adopted.

The remainder of this chapter draws on the observation in Chapter 3 of the importance of source modelling to compression performance, to develop a model of difference frames. The chapter develops the argument that in the case of natural video scenes, local motion will give rise to clustering of significant coefficients. It is then proven that source partitioning with optimal bit allocation will provide RD performance advantages. This proof is analytic and hence based on Gaussian assumptions.

In summary, this chapter argues that difference frame coding with source partitioning and RD optimization will provide a computationally feasible alternative to ME/MC. However, the RD performance of this system is as yet unknown.

8.1.5 Chapter 6 – Proposed Algorithm

The chapter details the design of the algorithm based on the premise of Chapter 5, as well as various methods drawn from the literature.

Chapter 5 indicates the potential of difference frames with optimal bit allocation for temporal decorrelation. However, the argument is based on assuming the source to be Gaussian, which is clearly an oversimplification. This chapter adopts the ρ-domain RD estimation algorithm, and Lagrange optimization techniques presented in Chapter 2, to provide a numerical solution to this source estimation and bit allocation problem.

In addition, the various still image compression routines reviewed in Chapter 4 are examined for suitability and the SPIHT algorithm chosen as the basis for the spatial decorrelation.

Finally an arithmetic coder is chosen as the entropy coder, based on prior work in this project.

The remainder of the chapter is dedicated to implementation details of this algorithm. The ρ-domain method has not previously been applied to wavelet video compression, thus its suitability had to be tested and several changes made.

This chapter explains how the theory of previous chapters was converted into a unified algorithm and implemented.

8.1.6 Chapter 7 – Results

This chapter presents the RD performance and computational complexity of the proposed algorithm.

Initially the proposed algorithm is compared to the standards. It is found that the proposed algorithm performs similarly to, or better than the standard methods. From this it is suggested that the RD performance loss of not using ME/MC, is counteracted by the use of wavelet spatial compression methods. In addition the algorithm is found to have a complexity an order of magnitude less than these methods.

The proposed algorithm is then compared to the wavelet video compression methods presented in Chapter 4. It is found that these methods outperform the proposed algorithm. In these cases, there is little reason to expect the proposed algorithm to be competitive, as the omission of either ME/MC or 3D coding is not countered by an advantageous spatial compression method, as the wavelet transform is used in all cases. In addition the optimal bit allocation of difference frames is inferior to complete ME/MC. Although complexity results are not published, it is reasoned in this chapter that the wavelet coders from the literature will have complexity similar to, or greater than the video standards presented previously. Thus our algorithm holds significant computational advantage over these methods.

This chapter then explores the RD performance / complexity trade-off of each block in the system. It is found that SPIHT accounts for over half the execution time. This suggests that either this algorithm may be improved or replaced in future work to gain maximum complexity reduction.

Examining the contributions of each block, for the Hallmonitor sequence, it is found that the RD estimation and optimal bit allocation account for 10% of the system complexity and contribute almost 20% of the performance.

The proposed temporal decorrelation method is then compared to the best method in the literature, using the Hallmonitor sequence at 25 kbps. It is found that the proposed method of difference frame coding through partitioning and optimal bit allocation offers a 185% PSNR increase, with a 10% execution time increase. The best method from the literature offers a 271% percent performance increase for a 400% execution time increase. The relative advantage of the proposed scheme is apparent.

Thus for limited local motion scenes, even at low bit rates (< 25 kbps), the proposed system offers acceptable (> 30 dB PSNR) visual quality and enormous computational advantage.

For scenes with distributed local motion(s), or global motion, although the proposed method offers computational advantage, it is unable to produce video of a sufficiently high fidelity at low bit rates for practical use. Thus the proposed algorithm is unsuited for such applications.

8.2 Discussion

The philosophy of this algorithm is to shift the computational effort from the temporal decorrelation stage, to the spatial decorrelation stage. This is done under the assumption that for low complexity, low rate, video coding the marginal rate of return on computation will be greater for this arrangement. This assumption is based on a model developed for difference frames.

This assumption is supported by the finding that block based ME/MC consumes a great majority of the computational resources in the standard methods of H.263 and MPEG. Thus it may be expected to consume similar resources in wavelet based methods, whether the ME/MC is performed in the spatial or wavelet domain. The only other proposed method of temporal decorrelation is 3D subband coding, however the memory requirements of this technique are extremely burdensome. Thus the only option for low complexity temporal decorrelation is difference frame coding.

Abandoning ME/MC will decrease the achievable RD performance. In order to balance this simplified temporal decorrelation, a sophisticated spatial decorrelation is adopted. Naturally this spatial decorrelator must exhibit good RD performance, in conjunction with low computational overhead. Fortunately the confluence of several factors permitted such an algorithm to be developed.

The first factor is that difference frames exhibit strong local stationarity, due to the underlying nature of video. Difference frames produce significant coefficients only for motion (in our case), and this motion must be localized, as the bodies which cause it are localized. This model of strong localization of significant coefficients is the basis of the algorithm. By partitioning the difference frames and adopting an optimal bit allocation scheme, this localization may be exploited for RD performance gain.

The second factor is a new RD estimation method. RD estimation, for optimal bit allocation, is generally a difficult problem. Chapter 2 demonstrates a remarkable RD estimation technique that is both fast and accurate. This method is the core enabling technique in our algorithm, as it is the basis of the optimal bit allocation. Previous RD estimation methods have been either slow or inaccurate, thus hindering optimal bit allocation.

As expressed in Chapter 6 this RD estimation drives a fast bit allocation scheme, which in turn controls the wavelet transform based SPIHT compression routine. All of these functions are performed by fast implementations, as testified by the execution time of the algorithm which is approximately an order of magnitude faster than the competing standards, and methods published in the literature.

8.3 Future Work

Future work on this algorithm is possible on several fronts.

Certain simple additions to the existing algorithm will yield good returns. Specifically, Chapter 7 notes that both frame rate and inter frame bit rate control will be valuable. As both of these functions may be built around an RD estimation and as such an estimate is already included in the algorithm, these additions should be trivial.

The spatial syntax provided by tiling is rudimentary. Chapter 7 indicates that although tiling is fundamental to the algorithm, tiling does decrease the compression achievable in the spatial coding stage. Thus, an adaptive tiling system, that automatically identifies areas of high coefficient clustering and is able to construct a tiling framework that optimizes the tile size and layout syntax as well as the overhead required to communicating the framework will probably be valuable.

The use of SPIHT for spatial decorrelation may be in error. Although Chapter 6 justifies the use of this algorithm from a theoretical and practical basis, better options are possible. Specifically the use of a context adaptive arithmetic coder will probably yield both RD and performance gains. Such an algorithm may be based on the ECECOW (Section 3.3.8) still image compression algorithm. The ECECOW algorithm successfully models the behaviour of still images in the wavelet domain and developing similar models of difference frames would allow the application of this algorithm in our system. This will probably yield RD performance advantages over SPIHT which is not constructed around a difference model.

Finally, the inclusion of a global motion estimation compensation strategy would expand the area of application of this algorithm. However, such algorithms are generally computationally expensive, thus voiding the fundamental advantage of this system. The design of low complexity global motion estimation / compensation system would have very widespread application.

8.4 Conclusion

This algorithm addresses the problem of producing video capable, mobile devices, running on standard processors.

Although the bandwidth of recent wireless channels is able to support low to medium bit rate video, the computational and memory insufficiency of most mobile devices has prevented advantage being taken of this opportunity. The price, size and additional power consumption of custom video compression chips has prevented their integration into embedded and mobile devices. Thus a need exists for a simple video compression system capable of execution on a standard DSP, and this work proposes such an algorithm.

This algorithm has an execution time approximately an order of magnitude less than the video both the current compression standards and other methods proposed in the literature, as well as modest memory requirements. The algorithm is capable of producing an acceptable and competitive quality video stream for low bitrate coding in cases of local motion.

Although it may be argued that mobile devices such as camera enabled cell phones may produce scenes with global motion, this is avoidable by careful use. The initial intended application of this algorithm is in equipment for use by trained personnel, hence the assumptions made by the algorithm can be guaranteed.

The main theoretical contribution of this work is the notion that correctly coded difference frames offer certain advantages for video coding. Specifically, the use and modification of the ρ-domain RD estimation method for wavelet video coding is unique to this work. However, this work is fundamentally practical, and integrates many existing ideas and methods into a single system with practical benefit.

There exists scope for future work on the algorithm, particularly with regard to better modelling of difference frames and better spatial syntax.

This project presents an engineering solution to an existing problem: current portable devices are unable to compress video due to the complexity of these algorithms, thus a low complexity algorithm is proposed. Although the performance of this algorithm is not a good as many other algorithms, this is irrelevant as the utility of having any video on a mobile is great. Furthermore most mobile devices are unable to display high quality video anyway. Thus this algorithm is designed specifically to address the core issue of video coder complexity and is highly successful in this regard.

[Anto92] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, *"Image Coding Using Wavelet Transform"*, IEEE Trans. Image Proc., Vol. 1, No. 2, pp.205-220, April 1992.

[Berg98] T. Berger, J. Gibson, *"Lossy Source Coding,"* IEEE Trans. On Info. Theory, Vol. 44, No. 6, Oct 1998.

[Bhas97] V. Bhaskaran, K Konstantinides, *"Image and Video Compression Standards,"* Kluwer Academic Publishers, 1997

[Boli00] M. Boliek, C. Christopoulos, E Majani, *"JPEG2000 Part I Final Draft International Standard, ISO/IEC FDIS15444-"*, ISO/IEC JTC1/SC29/WG1 N1855, Aug 2000.

[Chap98] S. Chapra, R. Canale, *"Numerical Methods for Engineers,"* Mc-Graw Hill, 1998

[Chri00] C. Christopoulos, A. Skodras, T. Ebrahimi, *"The JPEG2000 Still Image Coding Sytem: An Overview,"* IEEE Trans. Consumer. Electronics, Vol 46, No. 4, Nov 2000

[Chui92] C. K. Chui, *"An Introduction to Wavelets,"* Academic Press Inc, 1992

[Corb96] J. Ribas-Corbera, D Neuhoff, *"On the Optimal Motion Vector Accuracy for block-based motion-compensated video coders,"* Proc IS&T/SPIE Dig. Video Comp:Alg & Tech, San Jose, Feb 1996

[Corb99] J. Ribas-Corbera, S. Lei, *"Rate Control in DCT Video Coding for Low-Delay Communications,"* IEEE Trans. Circ and Syst. Vid. Tech. Feb 1999

[Daub92] I. Daubechies, *"Ten Lectures on Wavelets,"* CBMS-NSF, 1992

[Dono98] D. Donoho, M. Vetterli, R De Vore, I Daubechies, *"Data Compression and Harmonic Analysis,"* IEEE Trans. Info. Theory, Vol. 44, No. 6, Oct 1998

[Drag01] Dragotti, Vetterli , *"Wavelet Footprints: Theory, Algorithms, Applications"*, IEEE Trans Signal Proc., Submitted December, 2001

[Effe98] W. Effelsberg, R. Steinmetz, *"Video Compression Techniques,"* dpunkt-Verlag fur digitale Technologie, 1998.

[Forn84] G. D. Forney Jr., *"The Viterbi Algorithm,"* Proc IEEE, vol. 61, pp. 169-176, Feb 1984.

[Goya01] V. Goyal, "Theoretical Foundations of Transform Coding," IEEE Sig. Proc. Mag, Sept 2001.

[H263] M. Gallant, G. Cote, B. Errol, *"H.263+ TMN2.0 Reference Software,"* University of British Colombia, Canada, http://www.ee.ubc.ac.ca/image, 1997

[H263,2] ITU- Telecommunications Standarization Sector, Study Group 16 (T/SG16), *"Video Codec Test Model, Near Term Version 8 (TMN8),"* ITU, Portland, June 1997

[Hang97] H. Hang, J. Chen, "Source Model for Transform Video Coder and Its Application – Part 1: Fundamental Theory," IEEE Trans. Circ. And Syst. for Vid. Tech., Vol 7., No. 2, Apr 1997.

[He01] Z. He and S.K. Mitra, *"A Unified Rate-Distortion Analysis Framework for Transform Coding,"* IEEE Trans. Circ. Syst. For Video Tech, Dec 2001.

[He01,2] Z. He and S.K. Mitra, *"ρ-Domai Rate-Distortion Analysis and Rate Control for Visual Coding and Communication,"* Ph.D Thesis, University of California at Santa Barbara, April 2001

[Iyer01] L. Iyer, *"Image Compression Using Balanced Multiwavelets,"* MScEng Thesis, Virginia Polytechnic Institute and State University, June 2001

[Jack03] E. Jackson, R. Peplow, *"Fast Rate-Distortion Estimation and Optimization for Wavelet Video Coding,"* 3rd International Symposium on Image and Signal Processing and Analysis (IEEE), Rome, 2003

[Josh95] R. Joshi, V. Crump, T Fischer, *"Image Subband Coding Using Arithmetic Coded Trellis Coded Quantization,"* IEEE Trans. On Circ. And Syst. Vid. Tech., Vol 5, No 6, Dec 1995

[JPEG] "Digital Compression and Coding of Continuous-Tone Still Images," ISO/IEC IS 10918-1

[Kasn99] J. Kasner, M. Marcellin, B. Hunt, *"Universal Trellis Coded Quantisation,"* IEEE Trans. on Image Proc., Vol. 8, No. 12, Dec 1999

[Kim97] B. Kim and W. Pearlman, "An Embedded Wavelet Video Coder Using Three-Dimensional Set Partitioning in Hierarchical Trees," Proc IEEE Data. Comp. Conf, Mar 1997

[Kim00] B. Kim, Z, Xiong, W. Pearlman, *"Low Bit-Rate Scalable Video Coding with 3D Set Partitioning in Hierarchical Trees (3D-SPIHT),"* IEEE Trans. On Circ. And Syst. for Vid. Proc., Vol. 10, No. 8, Dec 2000

[Lee00] H. Lee, T. Chiang, Y-Q Zhang, *"Scalable Rate Control for MPEG-4 Video,"* IEEE Trans. Circ. And Syst. Vid. Tech, Vol. 10, No. 6, Sept 2000

[Lin01] K. K. Lin and R. M. Gray, *"Video Residual Coding Using SPIHT and Dependant Optimization,"* IEEE Proc. DCC 2001, pp 113-122

[Lin02] K K. Lin and R M. Gray, *"Rate Distortion Optimisation for the SPIHT Encoder, ". IEE*E 2002 Data Compression Conference;

[Lopr97] S. LoPresto, K. Ramchandran, M Orchard, "Image Coding based on Mixture Modeling of Wavelet Coefficients and a Fast Estimation Quantisation Framework," Proc 1997 Data Comp. Conf., 241-250, Mar 1997.

[JPEG2000,1] Testing Ad Hoc Group, *"JPEG testing results,"* ISO/IEC JTCI/SC29/WG1 N390, 1996

[Mall89] S. G. Mallat, "Multiresolution Approximations and Wavelet Orthonormal Bases of L2(R)," Trans. Amer. Math. Soc. Vol. 315(1), 1989

[Mall89,2] S. G. Mallat, *"Multifrequency Channel Decompositions of Images and Wavelet Models,"* IEEE Trans. Acoust. Speech Signal Proc. Vol 37(12), 1989

[Marp99] D. Marpe, H. Cycon, *"Very Low Bit-Rate Video Coding Using Wavelet Based Techniques,"* IEEE Trans. Circ. And Syst. for Vid. Tech. Vol. 9, No. 1, Feb 1999

[Mcin02] I. J McIntosh "Implementation of an Application Specific Low Bit Rate Video Compression Scheme," MScEng Thesis, Univ. Natal, Feb 2002.

[Marc90] M. Marcellin, T. Fischer, *"Trellis Coded Quantization of Memoryless and Gauss-Markov Sources,"* IEEE Trans. Comms. Vol 38., No. 1, Jan 1990

[Mart01] M. Martin, A.Bell, "New Image Compression Techniques Using Multiwavelets and Multiwavelet Packets," IEEE Trans of Image Proc., Vol. 10, No. 4, April 2001

[Marpe99] D. Marpe and H. L. Cycon, *"Very Low Bit-Rate Video Coding Using Wavelet-Based Techniques,"* IEEE Trans on Circ. And Syst. for Video Tech, Vol. 9, No. 1, Feb 1999

[MPEG2] *"MPEG-2 Encoder / Decoder, Version 1.2,"* MPEG Software Simulation Group, http://www.mpeg.org/MSSG/, July 19, 1996.

[MPEG4] *"(MPEG-4) Video Reference Software Version: Microsoft-FPDAM1-1.0-000703 Version 2"* ISO/IEC IS 14496-2 PDAM 1.0-000703, July 2000, http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/14496-5_Compressed_directories/Visual/

[MPEG4,2] Video Group, "Information technology -- Coding of audio-visual objects -- Part 1: Systems," ISO/IEC IS 14496-1:2001

[MPEG4,3] Video Group, "Information technology -- Coding of audio-visual objects -- Part 2: Visual, " ISO/IEC IS 14496-2:2001

[MPEG91] "Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5Mbit/s," ISO/IEC IS 11172, 1991

[MPEG93] *"MPEG-2 Video Test Model 5,"* ISO/IEC JTC1/SC29/WG11 MPEG93/457, April 1993.

[MPEG93,2] "Generic Coding of Moving Pictures and Associated Audio," ISO/IEC IS 13818

[Orte98] A. Orthega and K. Ramchandran, *"Rate-Distortion Methods for Image and Video Compression: an Overview,"* IEEE Sig. Proc. Mag., Nov 1998, No. 15(6), pp 23-50

[Papo91] A. Papoulis, "Probability, Random Variables, and Stochastic Processes," McGraw-Hill Book Co. 1991

[Penn88] W. Pennebaker, J. Mitchell, G. Langdon, R. Arps, *"An onverview of the basic principles of the Q-coder adaptive binary arithmetic coder,"* IBM Jour. Res. and Dev., 32(6), 717-726, 1988.

[Proa00] J. Proakis, *"Digital Communications,"* McGrawsHill Book Co. 2000

[Ram93] K. Ramchandran and M. Vetteli, *"Best Wavelet Packet Bases in a Rate-Distortion Sense,"* IEEE Trans. Image Proc, Vol. 2, No.2 April 1993

[Saha99] S. Saha and Rao Vemuri, *"Adaptive Wavelet Filters in Image Coders – How Important are They?,"* Proc. IEEE/IECON '99, Nov 1999, vol. 2, pp. 559-564.

[Shan48] C. Shannon, *"A Mathematical Theory of Communication,"* Bell Syst Tech. J.vol 27, July and Oct 1948

[Shan59] C. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in IRE Conv. Rec., vol. 7, 1959, pp. 142–163.

[Shap93] J . Shapiro, *"Embedded image coding using zerotrees of wavelet coefficients,"* IEEE Trans. Signal Processing, vol. 41, pp. 3445–3462, Dec. 1993.

[Shen99] K. Shen, E. Delp, *"Wavelet Based Rate Scalable Video Compression,"* IEEE Trans. On Circ. and Syst. for Vid. Proc., Vol. 9, No. 1, Feb 1999.

[Said96] A. Said and W. A. Pearlman, "*A New, fast and efficient image codecbased on set partitioning in hierarchical trees,*" IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 243–250, June 1996.

[Sola97] S. Solari, "Digital Video and Audio Compression," McGraw Hill, 1997

[Taub94] D. Taubman, and A. Zakhor, *"Multirate 3-D subband coding of video,"* IEEE Trans. Image Processing, Vol. 3, No. 5, Sep. 1994.

[Taub00] D. Taubman, *"High Performance Scalable Image Compression with EBCOT,"* IEEE Trans. Image Proc., Vol. 9, No. 7, July 2000.

[Thom96] G.B Thomas, R. L Finney, *"Calculus and Analytic Geometry,"* Addison-Wesley Publishing Company, 1996.

[Tsai96] M.J. Tsai, J. Villasenor, and F. Chen, *"Stack-run image coding,"* IEEE Transactions on Circuits and Systems for Video technology, vol. 6, pp. 519-521, October 1996

[UCLA1] UCLA Image Commun. Lab.,*"Wavelet Image Coding: PSNR Results,"* http://www.icsl.ucla.edu/~ipl/psnr_results.html.

[Vett01] M. Vetterli, *"Wavelets, Approximation and Compression,"* IEEE Sig. Proc. Mag, Vol 18, No 5, Sept. 2001

[Wall92] G. Wallace, *"The JPEG Still Picture Compression Standard,"* IEEE Trans. Consumer Electronics, Vol. 38, No. 1, Feb 1992

[Wu97] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," Proc. Of 31st Asilomar Conf. on Signals, Systems and Computers, 1378-1382, 1997

[Wu01] X Wu "Compression of Wavelet Transform Coefficients, The Transform and Data Compression Handbook," Ed. K. R. Rao et al. Boca Raton, CRC Press LLC, 2001

[Wang02] A.Wang, Z. Xiong, P. A. Chouy, and S. Mehrotray, *"Three-Dimensional Wavelet Coding of Video with Global Motion Compensation"* ,IEEE Data Compression Conference, 2002.

[Xiong96] Z. Xiong, K. Ramchandran and M. T. Orchard, ``*Wavelet packets image coding using space-frequency quantization,*" submitted to IEEE Trans. Image Processing, January 1996

[Xiong97] Z. Xiong, K. Ramchandran, and M. T. Orchard, *"Space-frequency Quantization for Wavelet Image Coding,"* IEEE, Trans, Image Processing, vol.6, pp. 677-693, May 1997.

[Xiong99] Z. Xiong, K. Ramchandran, M.Orchard, and Zhang, *"A Comparative Study of DCT- and Wavelet-Based Image Coding"* IEEE Trans on Circ. And Syst. for Vid. Tech. Vol 9, No. 5, August 1999

[Xiong99,2] Z. Xiong, X. Wu, "Wavelet Image Coding Using Trellis Coded Space-Frequency Quantization," IEEE Sig, Proc Letters, Vol. 6 No. 7, July 1999

[Xu01] Jizheng Xu, Shipeng Li, Ya-Qin Zhang and Zixiang Xiong. *"A Wavelet Video Coder Using Three Dimensional Embedded Subband Coding With Optimised Truncation (3-D ESCOT)"*, J. Appl. Computat. Harmonic Anal., vol. 10, pp. 290–315, May 2001.

[Xu02] Jizheng Xu, Zixiang Xiong, Shipeng Li, and Ya-Qin Zhang, *"Memory-Constrained 3-D Wavelet Transform for Video Coding Without Boundary Effects"*, IEEE Trans. On Circuits and Systems for Video Technology, Vol 12, No 9 September 2002

[Yang00] X. Yang and K. Ramchandran, "Scalable Wavelet Video Coding Using Aliasing Reduced Hierarchical Motion Compensation," IEEE Trons, On Image Proc. Vol 9, No 5 May 2000, Page 778

[Young93] R.W. Young and N.G. Kingsbury, *"Frequency-domain motion estimation using a complex lapped transform,"* IEEE Trans. Image Processing, vol. 2, pp 2-17, Jan 1993

[Zan02] Jinwen Zan, M. Omair Ahmad, and M. N. S. Swamy, *"New Techniques for Multi-Resolution Motion Estimation"*, IEEE Trans On Circuits And Systems For. Video Tech. Vol 12, No. 9, September 2002, Page 793

Appendix A - Introduction to the Wavelet Transform

The wavelet transform is the product of work in several diverse areas of mathematics, physics and engineering.

The foundations for understanding the transform lie in the mathematical treatment of wavelets as bases for function spaces. In Section A.1 and following we introduce wavelets from this perspective.

The recent popularity of wavelets comes from the possibility of performing the wavelet transform using digital filtering techniques. Multiresolution Analysis provides the mechanism whereby the connection between the continuous wavelet transform and filtering may be made. This field is investigated from Section A.5.

A.1 Functional Analysis

The following discussion utilizes the familiar territory of the Fourier Transform to motivate the wavelet transform. In doing so it is based heavily on the works of Chui [Chui92] and Daubechies [Daub92]. This appendix may be seen as a summary of these works, with some extra discussion.

A.1.1 Bases and Projections

In Section 1.1.1 we to provide the mathematical terminology to understand operations in function spaces, and explore norms, projections and classification of basis sets.

A.1.2 Framework

Consider the space of square integrable functions, $f \in L^2(\mathbb{R})$:

$$\int_{-\infty}^{\infty} |f(x)|^2 \, dx < \infty \tag{9.1}$$

The inner product and norm in this space are described respectively by :

$$\begin{aligned} \langle f, g \rangle &:= \int_{-\infty}^{\infty} f(x)\overline{g(x)}dx \\ \|f\|_2 &:= \langle f, f \rangle^{0.5} \end{aligned} \tag{9.2}$$

The problem of finding a suitable basis set for a function space amounts to finding a set of functions, $\{\varphi_i\}$, such that

$$f = \sum_{i}^{N} c_i \varphi_i \tag{9.3}$$

,

where the series is said to converge in L^2($\mathbb{R}$) if

$$\lim_{N\to\infty}\left\| f - \sum_i^N c_i\varphi_i \right\| = 0 \tag{9.4}$$

In signal compression the goal is to find a set {φ$_i$} such that f may be approximated by

$$\hat{f} = \sum_i^{k<N} c_i\varphi_i \tag{9.5}$$

with the goal of making k as smal! as possible without causing the L^2 norm of the error,

$$E = \left\| f(x) - \hat{f}(x) \right\| = \left(\int_{-\infty}^{\infty} \left| f(x) - \hat{f}(x) \right| dx \right)^{\frac{1}{2}}, \tag{9.6}$$

to grow beyond some 'acceptable' limit.

A.1.3 Orthogonality

If the set {φ$_i$} satisfies

$$\langle \varphi_i, \varphi_j \rangle = k\delta_{i,j}, k \in \mathbb{R}, \tag{9.7}$$

where the Kronecker symbol, δ_{ij}, is defined as:

$$\delta_{m,n} := \begin{cases} 1, m = n \\ 0, m \neq n \end{cases}, \tag{9.8}$$

then the basis set is said to be orthogonal.

Furthermore, if for an orthogonal set {φ$_i$}, where for each $i \in \mathbb{Z}$,

$$\|\varphi_i\| = 1, \tag{9.9}$$

that set is described as orthonormal.

A.1.4 Projection

The approximation / expansion operation is written as

$$\hat{f}(x) = \sum_i \langle \phi_i, f \rangle \phi_i, \tag{9.10}$$

where the inner product term, $\langle \phi_i, f \rangle$, is called the projection of f onto the basis set ϕ_i.

A.2 The Fourier Transform

Having introduced the machinery of transforms, the Fourier transform is taken as an example to illustrate these concepts. Its limitations are explored, and this leads to the development of the Gabor and wavelet transforms.

A familiar projection transform is the Fourier Transform, which is defined as follows :

$$(Ff)(\omega) := \hat{f}(\omega) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx = \frac{1}{\sqrt{2\pi}} \left\langle f, e^{i\omega x} \right\rangle, \tag{9.11}$$

where the inversion formula is given by

$$f(x) := (Ff)^{\vee}(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} [(Ff)(\omega)] e^{i\omega x} d\omega = \int_{-\infty}^{\infty} \left\langle f, e^{-i\omega x} \right\rangle e^{i\omega x} d\omega \tag{9.12}$$

Interpreting equation (9.11) in terms of (9.10) indicates that $\left\{e^{i\omega x}\right\}$ is the basis set $\{\varphi_i\}$. That is, the Fourier Transform is no more than the projection of a function f onto this basis set, and the Inverse Fourier Transform is the expansion of f in terms of the basis set.

A.2.1 Examination of the Fourier Transform Basis Set

The Euler Formula relates this basis set to trigonometric functions, which aids their classification:

$$e^{i\omega x} = \cos(\omega x) + i \cdot \sin(\omega x). \tag{9.13}$$

Clearly,

$$\begin{aligned} \left\langle \varphi_i, \varphi_j \right\rangle &= \left\langle e^{i\omega_0 x}, e^{i\omega_1 x} \right\rangle \\ \left\langle \varphi_i, \varphi_j \right\rangle &= \int_{-\infty}^{\infty} e^{i(\omega_O - \omega_1)} dx = \begin{cases} 1, \omega_O = \omega_1 \\ 0, \omega_O \neq \omega_1 \end{cases} \end{aligned} \tag{9.14}$$

which confirms the orthogonality of the basis set.

An important property of the Fourier Transform is the Parseval's Identity:

$$\left\| \hat{f} \right\|_2^2 = \left\| f \right\|_2^2 \tag{9.15}$$

which means that the energy of the signal is preserved in the transformation. This important property has motivated the use of this transform in much of signal processing. This preservation of the norm will be further explored in the wavelet transform.

The support of a function refers to the extent of the region of its domain for which the function is nonzero. Clearly each function in the basis set, $\left\{e^{i\omega x}\right\}$, has infinite support as each is a sinusoid of a particular frequency. This attribute makes this transform unsuitable for representing discontinuities succinctly.

Another means of stating the above is that the Fourier Transform provides absolute resolution in the frequency domain, with no joint information of the original (spatial or temporal) domain. This may be seen from equation (9.11) as the function f is projected onto a basis that extends over infinity, as indicated by the infinite bounds of the integral. Thus the transform domain indicates the total projection of the function, with no information as to the locality where the projection is strong or not. This information is 'smeared out' by the integration.

A.3 The Gabor Transform

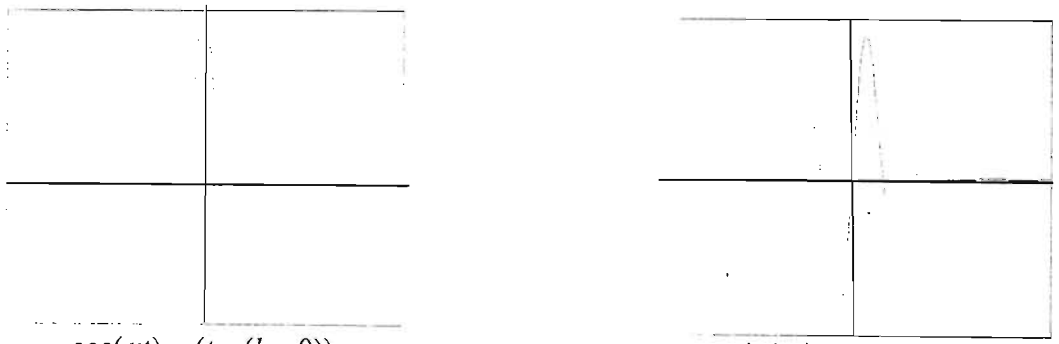
The first attempt to rectify the problem of infinite support of the Fourier basis functions was the Gabor Transform (also know as the Short Time Fourier Transform, STFT). In this transform the basis functions are multiplied by a Gaussian function,

$$g_\alpha(x) = \frac{1}{2\sqrt{\pi\alpha}} e^{\frac{-x^2}{4\alpha}} . \qquad (9.16)$$

Thus the transform is written as

$$(G_b^\alpha f)(\omega) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)\cdot(g_\alpha(x-b)\cdot e^{-i\omega x})dx = \frac{1}{\sqrt{2\pi}} \left\langle f, g_\alpha(x-b)\cdot e^{i\omega x} \right\rangle \qquad (9.17)$$

The purpose of this windowing is to produce basis functions that have compact support, as indicated by the example basis functions shown in Figure A-1.



$\cos(\omega t) g_\alpha(t-(b=0))$ $\qquad$ $\sin(\omega t) g_\alpha(t-(b=0))$

Figure A-1: Gabor Transform Basis Functions

This is done in order to allow the transform to produce both *approximate frequency and time/space* simultaneously. This represents a trade-off, as exact frequency information has been lost in order to obtain simultaneous, approximate time or space information.

The Gabor Transform of (9.17) then provides frequency information for a given choice of basis function, α , and translation, b. A complete transform then consists of taking (9.17) at sufficient values of b to cover the time duration of the signal f. That is, at a particular value of translation, b, the transform will give localized frequency information, ω. In order to obtain this information over the extent of the original signal, the basis must be translated over it, with b.

A.3.1 Examination of the Gabor Transform Basis Set

A.3.1.1 Completeness

The infinite integral of the Gaussian function, $g_\alpha(x)$ and its translates, $g_\alpha(x-b)$, is unity :

$$\int_{-\infty}^{\infty} g_\alpha(x-b)dx = \int_{-\infty}^{\infty} g_\alpha(x)dx = \int_{-\infty}^{\infty} \frac{1}{2\sqrt{\pi\alpha}} e^{\frac{-x^2}{4\alpha}} dx = 1 \qquad (9.18)$$

Thus

$$\int_{-\infty}^{\infty} \left(G_b^\alpha f\right)(\omega)\, db = (Ff)(\omega), \qquad (9.19)$$

that is, when taken over all real values of translation, the Gabor Transform provides the same frequency information as the Fourier Transform.

A.3.1.2 Time Frequency Windows

The manner in which the Gabor transform presents information is of importance. The joint frequency-time resolution may be represented on a grid, where the area of joint frequency-time resolution is called a window.

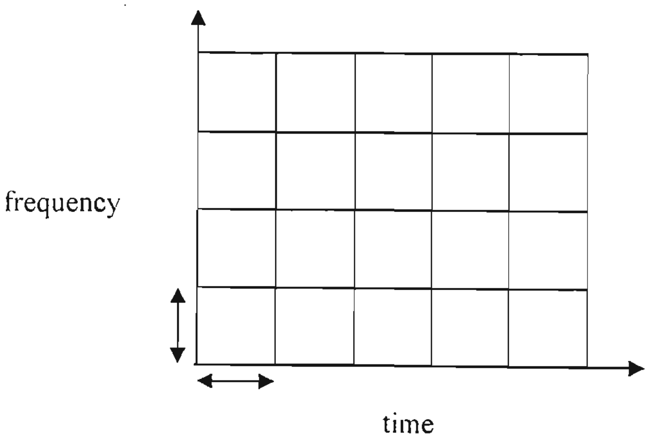


Figure A-2 : Time-Frequency Windows of the Gabor Transform

It must be noted that Heisenberg Uncertainty sets a minimum limit on the area (F*T) of the window. Thus in some sense the transform represents an optimisation. However, an improvement may be made my noting that the dimensions F and T are fixed in this transform.

This is undesirable as the time extent, T, may be insufficient to capture low frequencies, since a full period of such a signal would exceed T. A second difficulty lies in capturing short lived, high frequencies, such as is common in images (edges), as in this case T is too long to locate them accurately in T.

A.4 The Wavelet Transform

The wavelet transform represents a solution to these limitations of the Gabor Transform. Whereas the latter only has two parameters; α which controls the window area and b which sets the translation of the basis function, the former provides another, the scaling/dilation of the basis function. Considering a fixed basis function for now, in order to clarify this point, the Gabor basis generating function may be written as

$$G = G(t-b), \tag{9.20}$$

where b represents the translation of the basis function.

The wavelet basis function may be written

$$W = W\left(\frac{t-b}{a}\right). \tag{9.21}$$

The parameter, a, determines to what extent the function W is spread over time, ie its dilation.

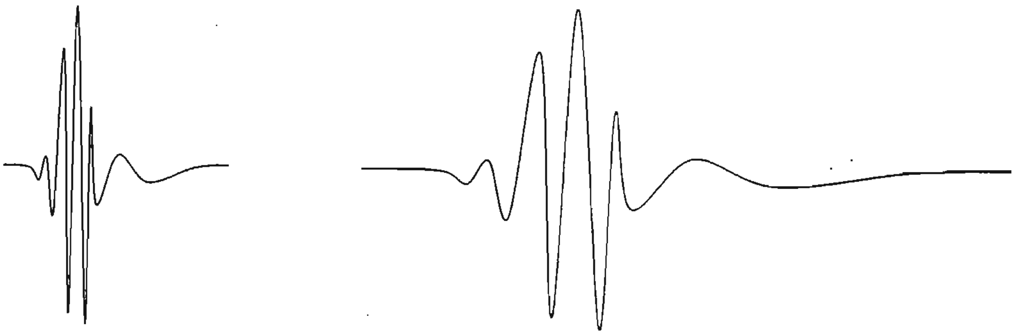


Figure A-3: Dilation of a Wavelet

The advantage of this is that the time frequency window is altered as illustrated in Figure A-4 below.

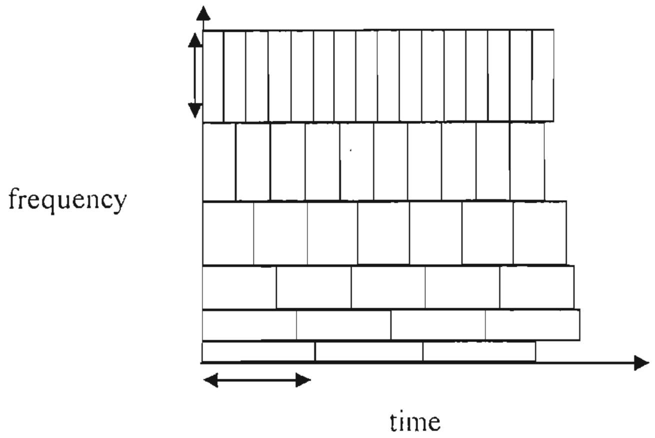


Figure A-4: Time Frequency Windows of the Wavelet Transform

Here it can be seen that at low frequencies T increases to allow the period of the function to be captured and at high frequencies T decreases to allow accurate location of these frequencies.

Incorporation of this new variable into the transform framework yields the following as the wavelet transform:

$$(W_{\psi}f)(a,b)=|a|^{1/2}\int_{-\infty}^{\infty}f(t)\overline{\psi\left(\frac{t-b}{a}\right)}dt, \tag{9.22}$$

where ψ is the as yet undefined wavelet basis function. As a projection the transform may be written as :

$$(W_{\psi}.f)(b,a)=\left\langle f,a^{1/2}\psi\left(\frac{t-b}{a}\right)\right\rangle. \tag{9.23}$$

A.4.1 Examination of the Wavelet Transform Basis Set

In order to qualify as a wavelet the function , ψ must satisfy the Admissibility Criterion :

$$C_{\psi}=\int_{-\infty}^{\infty}f(t)\psi\left(\frac{|\hat{\psi}(\omega)|^{2}}{|\omega|}\right)d\omega<\infty. \tag{9.24}$$

As can be seen, this requires the function to have a bandpass spectrum, hence be of compact support. This is the origin of the term 'wavelet.'

A.4.1.1 Dyadic Wavelets

Many choices are possible for the wavelet function but the most useful formulation is

$$a^{1/2}\psi\left(\frac{t-b}{a}\right)=2^{j/2}\psi\left(2^{j}t-k\right):=\psi_{j,k}(x) \tag{9.25}$$

which allows the wavelet transform to be redefined

$$\left(W_{\psi}f\right)(b,a)=\left\langle f,\psi_{j,k}(t)\right\rangle. \tag{9.26}$$

This construction is known as the Dyadic Wavelet Transform and it has several properties that facilitate its use without excessive mathematical design. Most importantly, under certain achievable restrictions, it is possible to obtain the dual readily.

A.4.1.2 Duals

The following discussion is based on [Chui92], full proofs are to be found there. A transform is considered to be norm bounded, in an l_2 sense, if for the sequence $\{c_{j,k}\}$:

$$\left\|\left\{c_{j,k}\right\}\right\|_{l2}^{2}:=\sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}\left|c_{j,k}\right|^{2}<\infty \tag{9.27}$$

the projection

$$A\left\|\left\{c_{j,k}\right\}\right\|_{l2}^{2}\leq\left\|\sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}c_{j,k}\psi_{j,k}\right\|_{2}^{2}\leq B\left\|\left\{c_{j,k}\right\}\right\|_{l2}^{2} \tag{9.28}$$

for $0<A\leq B<\infty$. Essentially this means that the norm does not vanish or expand to infinity in the transform domain. This is required in order to recover the sequence from this domain.

If $\psi_{j,k}$ is orthonormal in terms of the projection of an $f\in L^2(\mathbb{R})$ onto $\psi_{j,k}$, converges in $L^2(\mathbb{R})$ and is norm bounded, then it is said to form a Riesz basis of $L^2(\mathbb{R})$. Such a function is termed an $\Re$-function.

The relevance of this is that, if $\psi_{j,k}$ is an $\Re$-function, then there is a unique basis $\left\{\psi^{j,k}\right\}$ which is also a Riesz basis of $L^2(\mathbb{R})$ and is 'dual' to $\psi_{j,k}$:

$$\left\langle\psi_{j,k},\psi^{k,m}\right\rangle=\delta_{j,l}\cdot\delta_{k,m} \tag{9.29}$$

which allows

$$f=\sum_{j,k=-\infty}^{\infty}\left\langle f,\psi_{j,k}\right\rangle\psi^{j,k} \tag{9.30}$$

This is important as it guarantees that the projection is reversible. It must be noted that if $\{\psi_{j,k}\}$ is an orthonormal basis, then $\psi_{j,k}\equiv\psi^{j,k}$. In fact the following section on multiresolution analysis is conducted under this assumption.

A.5 Multiresolution Analysis

The theory of multiresolution provides the connection between the treatment of wavelets in the purely mathematical domain and their implementation by engineers as iterated filters banks.

This is the theory that provides credibility to the design and analysis of wavelets using digital filtering techniques and is thus crucial.

A.5.1 The Multiresolution Subspace Construction

The principle is to split the $L^2(\mathbb{R})$ space into a hierarchy of embedded closed subspaces, V_j, where

$$\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots, \tag{9.31}$$

and

$$\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R}) \tag{9.32}$$

and

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\} \tag{9.33}$$

In addition to the properties stated above, the subspaces are required to be closed under translation and be scaled versions of the V_0 space :

$$f \in V_0 \Rightarrow f(\cdot - n) \in V_0, \forall n \in Z \tag{9.34}$$

and

$$f \in V_j \Leftrightarrow f(2^j \cdot) \in V_0 \tag{9.35}$$

The operation of orthogonal projection of a function, $f \in L^2(\mathbb{R})$, onto a subspace, V_j ,is denoted by $P_{j-1}f$. In the case of the subspaces shown above projection onto subspaces of successively smaller index captures increasingly more detail.

A.5.2 Multiresolution

The multiresolution argument is that for every subspace, V_j, there exists an orthogonal subspace W_j, such that :

$$V_{j-1} = V_j \oplus W_j. \tag{9.36}$$

Loosely the subspace W_j contains the 'details' required to move from V_j to V_{j-1}.

Furthermore the subspaces W_j are mutually orthogonal, ie

$$W_j \perp W_k, k \neq j. \tag{9.37}$$

Thus, any subspace V_J may be constructed from the orthogonal summation from a lower detail subspace V_j and the intervening W_j as

$$V_J = V_j \oplus \bigoplus_{k=j}^{J-1} W_k \tag{9.38}$$

where the summation is taken to be orthogonal. This implies that:

$$L^2(\mathbb{R}) = \bigoplus_{j\in Z} W_j , \tag{9.39}$$

The subspace hierarchy can thus be depicted as :

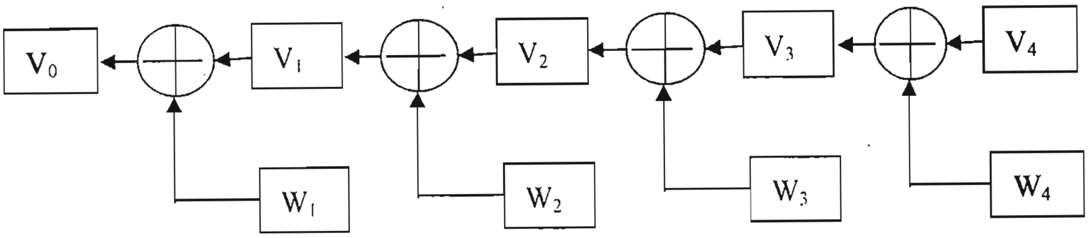


Figure A-5: Subspace Hierarchy

The utility of this operation is that it allows a function to be analysed at various levels of detail. For instance, in compression it is desirable to remove the details from a function, which is what a projection prom one level V_{j-1} to another, V_j, achieves. For the purpose of analysis, these details are retained in W_j, which is often useful.

A.5.3 The Scaling Function

Consider now a function, ϕ, such that its integral translates,

$$\{\phi_{0,n}; n\in\mathbb{Z}\}, \tag{9.40}$$

form an orthonormal basis of V_0.

For our purposes consider further that $\phi_{j,k}$ is of the form

$$\phi_{j,k} = 2^{-j/2}\varphi(2^{-j}\cdot - k) \tag{9.41}$$

Examining this form it is evident that the translation term, k, allows the function to form a basis within any given subspace. Furthermore the scaling term, 2^{-j}, allows the function to exist in any given subspace, j, and thus form a basis for any subspace j. Finally the term $2^{-j/2}$ is an energy normalisation across these scales. The function ϕ is called the scaling function.

A.5.4 The Wavelet Function

In a similar way a function ψ may be defined such that

$$\psi_{j,k} = 2^{-j/2}\psi(2^{-j}\cdot -k) \qquad (9.42)$$

where in this case the function forms an orthonormal basis for the W_j subspaces by the same argument as above.

A.5.5 Projections

Consider a function $f \in V_j$. This function may be written as

$$f = \sum_{k\in\mathbb{Z}} \langle f, \phi_{j,k} \rangle \phi_{j,k} \qquad (9.43)$$

since it has no component outside the subspace V_j. With the use of the same terminology equation (9.36) may be rewritten to describe a function $f \in L^2(\mathbb{R})$.

$$P_{j-1}f = \sum_{k\in Z} \langle f, \phi_{j,k} \rangle \phi_{j,k} + \sum_{k\in Z} \langle f, \psi_{j,k} \rangle \psi_{j,k}\,. \qquad (9.44)$$

Now consider the operation of projecting the basis function, φ onto the next finer level of detail. We have

$$\begin{aligned} &\phi \in V_0 \subset V_{-1} \\ &\therefore \phi = \sum_k \langle \phi, \phi_{-1,n} \rangle \phi_{-1,n} \\ &\therefore \phi = \sum_n h_n \cdot \phi_{-1,n} \end{aligned} \qquad (9.45)$$

and due to orthormality

$$\sum_{n\in\mathbb{Z}} |h_n|^2 = 1\,. \qquad (9.46)$$

Thus the scaling function may be written as a function of the scaling function at the next level of detail. This is entirely sensible, as a subband with greater detail can by definition represent a function contained in a coarser subband.

Similarly consider the wavelet function, ψ :

$$\begin{aligned} &\psi \in W_0 \subset V_{-1} \\ &\therefore \psi = \sum_k \langle \psi, \phi_{-1,n} \rangle \phi_{-1,n}, \\ &\therefore \psi = \sum_k g_n \cdot \phi_{-1,n} \end{aligned} \qquad (9.47)$$

where again

$$\sum_{n\in\mathbb{Z}} |g_n|^2 = 1 . \qquad (9.48)$$

It is apparent that both the scaling and wavelet functions are functions of the scaling function at the next level of detail. This precipitates the question of how h_n and g_n are related. Through a complicated proof utilising Fourier tactics, which is not reproduced here, Daubechies proved the following theorem [Daub92, pg 132-135] (with slight modifications for readability):

If a ladder of closed subspaces $(V_j)_{j\in Z}$ *in* $L^2(\mathbb{R})$ *satisfies equations* (9.31) *to* (9.35), *then there exists an associated orthonormal wavelet basis* $\{\psi_{j,k}; j,k \in \mathbb{Z}\}$ *for* $L^2(\mathbb{R})$ *such that*

$$P_{j-1}.f = P_j + \sum_{k\in Z} \langle \cdot, \psi_{j,k} \rangle \psi_{j,k} \qquad (9.49)$$

where

$$\begin{aligned}\psi &= \sum_n (-1)^{n-1} h_{-n-1} \phi_{-1,n} \\ \psi(x) &= \sqrt{2} \sum_n (-1)^{n-1} h_{-n-1} \phi(2x-n)\end{aligned} \qquad (9.50)$$

This is the heart of the multiresolution analysis. Both scaling and wavelet functions are related directly to the scaling function of the next higher level of detail.

A.5.6 Digital Filtering

It is clear that equations (9.45) and (9.47) may be seen to represent digital filters, but for completeness the formal development is given here.

From equation (9.47)

$$\psi = \sum_k g_n \cdot \phi_{-1,n} \qquad (9.51)$$

with

$$g_n = \langle \psi, \phi_{-1,n} \rangle = (-1)^n h_{-n+1} , \qquad (9.52)$$

it may be developed that

$$\begin{aligned}\psi_{j,k}(x) &= 2^{-j/2} \psi(2^{-j} x - k) \\ &= 2^{-j/2} \sum_n g_n 2^{1/2} \phi(2^{-j+1} x - 2k - n) \\ &= \sum_n g_n \cdot \phi_{j-1,2k+n}(x) \\ &= \sum_n g_{n-2k} \phi_{j-1,n}(x)\end{aligned} \qquad (9.53)$$

Thus

$$\left\langle f,\psi_{1,k}\right\rangle = \sum_{n} \overline{g_{n-2k}} \left\langle f,\phi_{0,n}\right\rangle \tag{9.54}$$

Thus

$$\left\langle f,\psi_{j,k}\right\rangle = \sum_{n} \overline{g_{n-2k}} \left\langle f,\phi_{j-1,n}\right\rangle \tag{9.55}$$

This may be viewed as the convolution of the sequence $\left\langle f,\phi_{j-1,n}\right\rangle$ with the sequence g_{n-2k}, which is equivalent to a digital filtering with coefficients, g_{n-2k}, followed by downsampling by two. This means that the projection of a function onto W at a coarser resolution may be found by filtering its projection on Vat the next finer level.

Similarly

$$\begin{aligned} \phi_{j,k}(x) &= 2^{-j/2}\phi(2^{-j}x-k) \\ \phi_{j,k}(x) &= \sum_{n} h_{n-2k}\phi_{j-1,n}(x) \end{aligned} \tag{9.56}$$

from which we can obtain the functions projection on V$_j$

$$\left\langle f,\phi_{j,k}\right\rangle = \sum_{n} \overline{h_{n-2k}} \left\langle f,\phi_{j-1,n}\right\rangle . \tag{9.57}$$

A.5.7 Summary of Multiresolution

In summary the $L^2(\mathbb{R})$ function space is segmented into a hierarchy of subspaces V_j and W_j, where an increase in index represents a lower resolution space. The difference in resolution between V_j and V_{j-1} is contained in W_j. In terms of mathematical notation, the projection relationship is written as

$$P_{V_{j-1}}f = P_{V_j}f \oplus P_{W_{j-1}}f . \tag{9.58}$$

In order to investigate this projection the basis sets for each subspace need to be defined. The basis set for the V_j subspaces is given by the scaling function set

$$\phi_{j,n}(x) = 2^{-j/2}\phi(2^{-j}x-n), \tag{9.59}$$

and the basis for the W_j subspaces is given by the wavelet function set

$$\psi_{j,n}(x) = 2^{-j/2}\psi(2^{-j}x-n), \tag{9.60}$$

that is, any $f \in V_j$ may be written as

$$f = \sum_{k\in\mathbb{Z}} \left\langle f,\phi_{j,k}\right\rangle \phi_{j,k} \tag{9.61}$$

,

where the inner product is the operation of projection onto the subspace V_j, A similar procedure holds for $f \in W_j$.

Multiresolution comes from the property that with basis sets defined as above, the following relationships hold:

$$P_{V_j} = \left\langle f, \phi_{j.k} \right\rangle = \sum_n \overline{h_{n-2k}} \left\langle f, \phi_{j-1.n} \right\rangle = \sum_n \overline{h_{n-2k}} P_{V_{j-1},n} \qquad (9.62)$$

and

$$P_{W_j} = \left\langle f, \psi_{j.k} \right\rangle = \sum_n \overline{g_{n-2k}} \left\langle f, \phi_{j-1.n} \right\rangle = \sum_n \overline{g_{n-2k}} P_{V_{j-1},n} \qquad (9.63)$$

The relevance of this is that the projection of f onto any of the subpaces in the hierarchy may be obtained by a digital filtering of its projection onto the next finer subspace.

A.6 The Wavelet Transform for Image Processing

A.6.1 The Iterated Filter Bank Arrangement

The previous sections of this appendix have briefly developed the wavelet transform from the perspective of functional analysis and then multiresolution analysis. This section will briefly explain how the wavelet transform of an image is taken.

In brief summary the wavelet transform of a single dimensional function, f, may be written in continuous form as

$$\left(W_{\psi}f\right)(a,b) = |a|^{1/2} \int_{-\infty}^{\infty} f(t) \overline{\psi\left(\frac{t-b}{a}\right)} dt. \qquad (9.64)$$

Equivalently, the multiresolution perspective allows the discrete wavelet transform to be defined as a series of projections. At each level, *j*, the subspaces are called V_j and W_j, and each level is simply a digitally filtered version of the projection onto V_{j+1}, by equations(9.62) and (9.63).

This may be depicted:

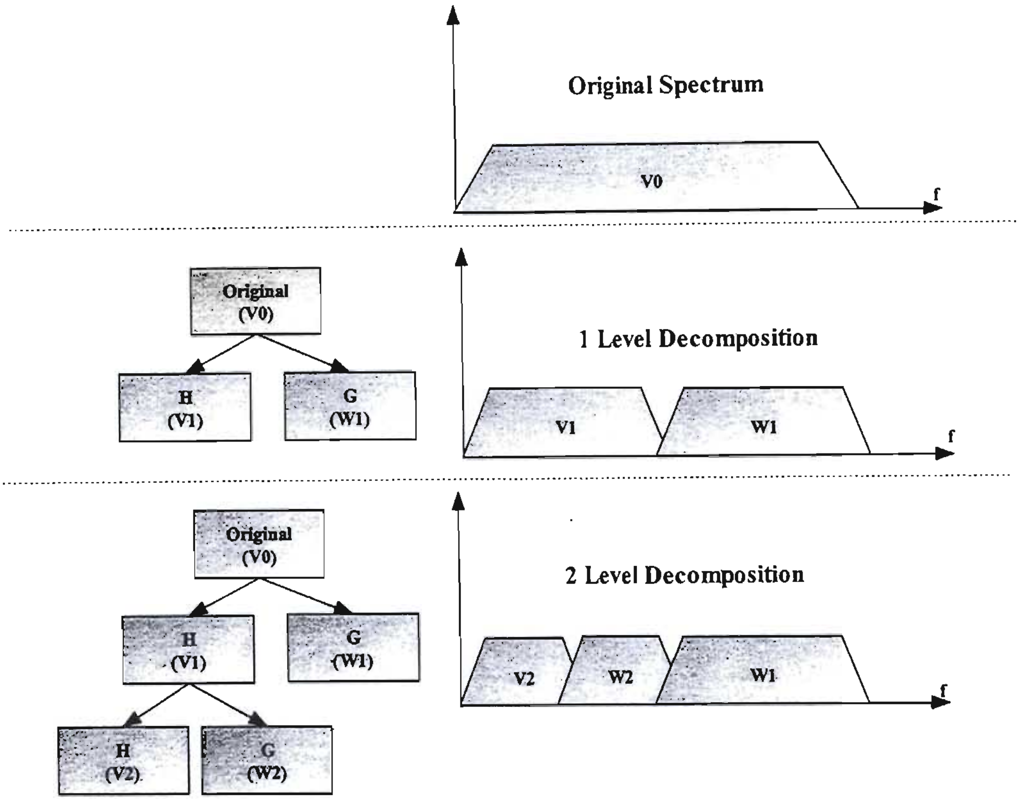


Figure A-6: Wavelet Filtering

Here the original spectrum falls into the Fourier subspace V_0. By applying the highpass decimating (by 2) digital filter G and lowpass digital filter H, this subband is split into two subbands, W_1 and V_1. By then filtering the lowpass band, V_1, using the same decimating filters as before, the subband, V_1, is further split into subbands W_2 and V_2. This may be continued until the data is exhausted.

This arrangement leads to the wavelet transform often being implemented as an iterated digital filter bank.

A.6.2 Two Dimensional Filtering

In order to process images that are 2D sources, the filter bank arrangement must be modified. The process is simply achieved by separating the horizontal and vertical filtering processes. The image is first filtering in one direction (horizontal or vertical) and then the coefficients output are filtered in the other direction (vertical or horizontal), the order of filtering has no effect.

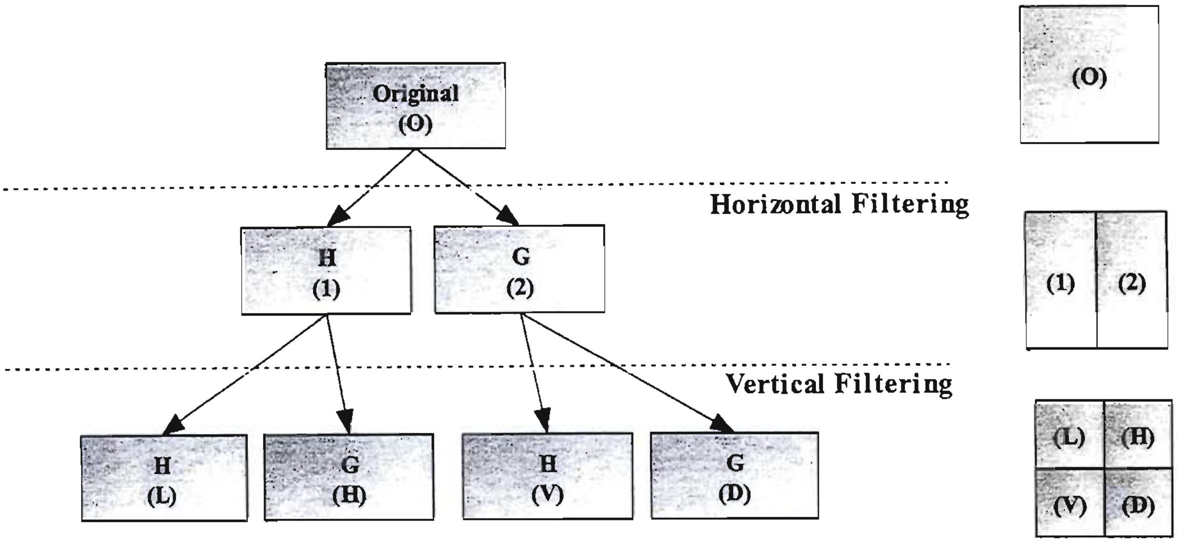


Figure A-7: 2D Filtering

Figure A-7 above illustrates the process. The original image is filtered by the H and G filters, in the horizontal direction, downsampling the output by two. The output is arranged as shown to right of Figure A-7. Then each of these coefficient sets is filtered by H and G in the vertical direction, decimating by two again. The output is arranged in the 4 blocks as shown. The (L) block has been low pass filtered in the horizontal and vertical direction and thus contains the average information of the frame. The block, (H), has been low pass filtered horizontally and high pass filtered vertically, and thus contains the horizontal details such as edges. Similarly (V) contains the vertical details and (D) the diagonal detail. This arrangement is illustrated below for the popular Lena image.

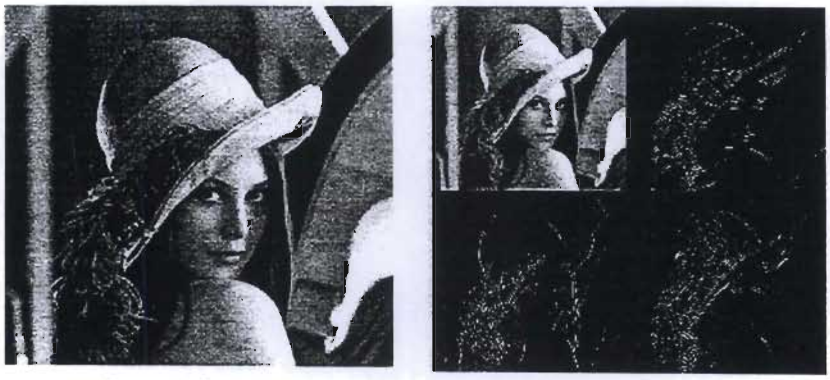


Figure A-8: Lena Image with Single Level Wavelet Transform

This arrangement constitutes a single level, two-dimensional wavelet transform. The filter iteration may be achieved as before by filtering the (L) through the same filter set. In this way a multiple level wavelet decomposition may be achieved.