PHD

Rate distortion control in digital video coding

Zhang, Haoxiang

Award date:
2007

Awarding institution:
University of Bath

RATE DISTORTION CONTROL IN DIGITAL VIDEO CODING

submitted by

Haoxiang Zhang

for the degree of

Doctor of Philosophy

of the

University of Bath

2007

Signature of the Author...

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

To

Fusheng and Jilan,
my parents,
for their constant love and care;

and Lina,
my love and companion,
for her encouragement and support

Abstract

Lossy compression is widely applied for coding visual information in applications such as entertainment in order to achieve a high compression ratio. In this case, the video quality worsens as the compression ratio increases. Rate control tries to use the bit budget properly so the visual distortion is minimized.

Rate control for H.264, the state-of-the-art hybrid video coder, is investigated. Based on the Rate-Distortion (R-D) slope analysis, an operational rate distortion optimization scheme for H.264 using Lagrangian multiplier method is proposed. The scheme tries to find the best path of quantization parameter (QP) options at each macroblock. The proposed scheme provides a smoother rate control that is able to cover a wider range of bit rates and for many sequences it outperforms the H.264 (JM92 version) rate control scheme in the sense of PSNR.

The Bath University Matching Pursuit (BUMP) project develops a new matching pursuit (MP) technique as an alternative to transform video coders. By combining MP with precision limited quantization (PLQ) and multi-pass embedded residual group encoder (MERGE), a very efficient coder is built that is able to produce an embedded bit stream, which is highly desirable for rate control. The problem of optimal bit allocation with a BUMP based video coder is investigated. An ad hoc scheme of simply limiting the maximum atom number shows an obvious performance improvement, which indicates a potential of efficiency improvement.

An in depth study on the bit Rate-Atom character has been carried out and a rate estimation model has been proposed. The model gives a theoretical description of how the bit number changes. An adaptive rate estimation algorithm has been proposed. Experiments show that the algorithm provides extremely high estimation accuracy.

The proposed R-D source model is then applied to bit allocation in the BUMP based video coder. An R-D slope unifying scheme was applied to optimize the performance of the coder. It adopts the R-D model and fits well within the BUMP coder. The optimization can be performed in a straightforward way. Experiments show that the proposed method greatly improved performance of BUMP video coder, and outperforms H.264 in low and medium bit rates by up to 2 dB.

Acknowledgements

Firstly, I would like to express my thanks to Prof. Donald M. Monro and Dr. Adrian N. Evans for their supervision of this work. Their continuous guidance, encouragement, and insightful advice have made a unique contribution to this project.

I would also like to thank my parents for their love and support. They never gave up their hope on me, however difficult the situation was, and incessantly supplied love and affection. This kept me going all these years.

Thanks are also given to my colleagues, Charith, Eric, Nick, Yannis, Simon, Yuan, Dexin, Soumyadip, Fei, for I have benefited a lot from the discussions with them; especially to Leo, and Rachel, for our cooperation on part of this project.

I would like to thank Universities UK and the University of Bath for financially supporting my PhD course with ORS and Bath Bursary.

My appreciations are also to all those who helped me, in both research and life. This thesis would not have been possible without your assistance.

Contents

List of Figures

List of Tables

Statement of Originalities

The author considers the following elements in this work form an original contribution to rate control in digital video coding literature.

Chapter 2

- The design and implementation of unified video reference framework;

Chapter 3

- Analysis of the features of the quantization scheme in H.264, and its challenge to rate control;
- The design and proposal of the slope unifying optimization scheme based on the rate distortion analysis;
- Investigation on the validity of the monotony of rate-distortion slope changes at macroblock level;
- Proposal of the Lagrangian multiplier method for quantization parameter picking scheme to optimize the coding performance;
- Comparison between the proposed scheme and the original H.264 rate control algorithm. Data have been obtained to reveal the improvement of the proposed method.

Chapter 4

- Proposal of adopting embedded frame coder into video coding system. Analyzed the benefit that the scalable bit stream brings to rate control;
- Application of two typical embedded schemes in video coder. Performed experiments and analysis about the proposed system's efficiency;
- Design and implementation of the hybrid video coder with H.264 and matching pursuit; proposed and applied the bit rate calculating method;

- Analysis of the general rate-distortion relationship in the BUMP video coder, and the proposal of the bit rate smoothing algorithm with atom limitation;
- Investigation of the optimal setting of the Maximum-Atoms and the Initial quantization parameter for different sequences at different bit rates in order to improve the system coding efficiency;
- Experiments and performance comparison of the tuned system and the original H.264 video coder.

Chapter 5

- Proposed the distortion-atom relationship model based on the matching pursuit theory;
- Analysis of the relationship between bits number and the atom number, and the character of the bits number changing curve;
- Proposal of a novel mathematical framework for bit number modelling, which is of a very simple form and shows very high accuracy;
- Design of an adaptive bits number prediction method based on the proposed bits modelling framework;
- Comparison of the predicted bits number and the actual bit number obtained during practical video coding,
- Analysis of the prediction error of the proposed bits estimation method.

Chapter 6

- Detailed analysis of the R-D slope unifying method and its applicability to the BUMP based video coder;
- Detailed analysis of the impact of different quantization parameter choices to the motion prediction bits cost and the residue frame distortion;
- Proposal of the quantization parameter calculation and prediction method for the hybrid video coder;

- Design of some specific features for the optimization method to be applied to video coder;
- Application of the slope unifying method to video coding, and comparison of the experimental results with the original H.264 rate control scheme.

Chapter 1 INTRODUCTION

In the last few decades, striking advances of technologies in networks and telecommunications have heralded a new era of audio-visual information communication. The advances have enabled users to exchange or share information using different media, such as voice, audio, video, text and images. Video is arguably the most popular media form that also has the largest bandwidth or storage requirements.

Until a few years ago, digital video and images could only be found in professional studios, or for some special uses. But now, they are widely available in various areas, such as home vision entertainment, medical and industry research, web pages, mobile phones, etc. They can also be conveyed by different ways, like the Internet and Wireless channels. Together with the development in communication systems, the evolution of compression techniques plays an important role in the popularization of digital video and image.

To increase the compression ratio of digital video and images, compression techniques often introduce visual quality reduction and loss of information compared with the original video sequences. This loss is also known as distortion. Normally, the distortion gets higher while the compression ratio is increased. To implement a decent application, a high compression ratio is required to achieve efficiency, and distortion needs to be kept low to maintain the video quality. Here arises the problem of rate-distortion control: to make the best compromise between the bandwidth or storage space and the ultimate video signal quality. This thesis investigates techniques to achieve the compression file at target rate, and on how to improve the video quality at a given rate.

In this chapter, the problem of rate control in digital video is addressed. Then the basic motivation and application of digital compression, and some widely used techniques are introduced. Finally, the contribution of the thesis is outlined.

1.1 The problem formulation

1.1.1 Motivation for compression

Digital signals are more and more widely used today due to some unique advantages. They can easily tap into the ever-increasing power of semiconductors, have got high resilience to interference, and they can be perfectly copied an infinite number of times. Finally, digital signals can be encrypted to provide some measure of protection.

However, because of the large amount of information conveyed by digital signals, storing digital signals often requires huge storing capacity. This problem also comes with digital video. A simple calculation tells that the data contained in even a very short video clip is huge.

For example, CCIR-601 is an International Telecommunication Union (ITU-T, formally CCITT) standard for image format for studio quality. CCIR-601/625 is for European broadcasting, which defines that the number of lines per frame is 625, each having 720 pixels and the number of frames per second is 25. The number of active lines per frame in CCIR-601/625 is 576, so the number of pixels contained in one second is equal to720x576x25=10368000 pixels. Considering that 8 bits are used for representing each pixel, the total bit rate becomes 10368000x8=82944000 bits/s. This is only to convey greyscale video. For colour video, this value needs to be tripled if RGB colour space is used. Considering the bandwidth for analogue video of PAL mode is only around 13.5MHz, such a number for digital representation is far too large and makes it necessary to introduce compression in practice.

The main objective of a video compression system is to represent a video sequence with as few bits as possible while preserving the level of image detail and quality required for the given application. The smaller the amount of data, the less storage and transmitting resources are required, enabling a higher transmitting speed and efficiency, and cheaper expenses of implementation. All these are necessary for practical applications.

Some might argue that with the rapid advances of communication technologies, such as use of optical fibre, the bandwidth might not be a problem. But due to the fact that the demand of digital video services is also growing exponentially, compression schemes would always be necessary. Additionally, with compression techniques, more services of better quality can be implemented with the limited available bit rate, and at much lower implementation expenses.

1.1.2 Rate control in video compression

1.1.2.1 Lossy compression—the advantage and drawback

Compression is about reducing the size of the original signals, and it is possible because of the existence of redundancies. Shannon [1] introduced the idea of entropy in his information theory, and defined the entropy to be a measure of information content. The entropy can be regarded as the pure information,which is defined as,

$$H(X)=\sum_{i=1}^{n}p(x_i)\log_2(\frac{1}{p(x_i)})=-\sum_{i=1}^{n}p(x_i)\log_2 p(x_i) \qquad (1.1)$$

H(X) is the average entropy of the source; x_i is one of the possible events from the source, and $p(x_i)$ represents the probability of that event .The concept of entropy allows the measurement of information and redundancies.

A normal video signal contains both pure information and redundancy. Video compression is achieved by removing the redundancy existing in the data. There are various approaches to video compression, and the common purpose among those video encoders is to exploit the statistical properties of the video data by using variable length code words to convey the information.

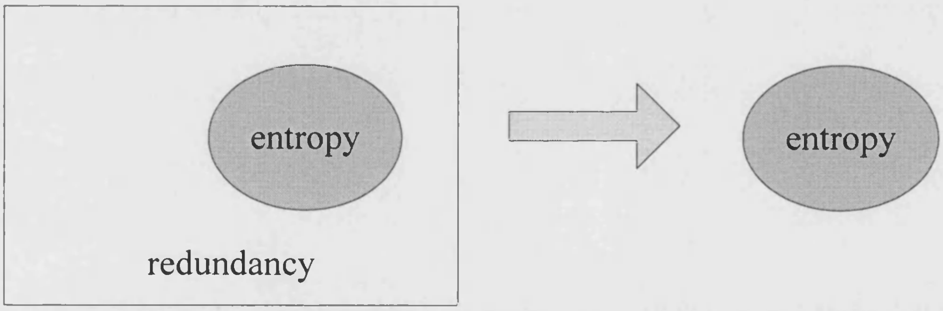


Figure 1.1 Complete removal of redundancy

Figure 1.1 shows an ideal situation of compression. The original data contains both “pure” information, and redundancies. After compression, the redundancies are completely removed, and the entropy is fully retained. Therefore, an exact replica of the original data can be constructed after decoding. This kind of compression is called

Lossless Compression. The information entropy also indicates the lower bound of the number of bits required for expressing that information. This number in turn limits the compression ratio that can be achieved by lossless coding. Therefore, the bit rate often cannot be reduced much if no information loss is allowed.

As introduced before, the prime objective of compression is to achieve the smallest possible size of data to meet the limited availability of resources, such as bandwidth and storage facility. The limited compression ratio of lossless coding often cannot fit the signal into the available bandwidth in many situations. On the other hand, it is quite often that some irrelevant information can be discarded together with those redundancies during compression, and some reasonable information fidelity is still maintained.

Actually, in many applications, such as entertainment, the most accepted solution is to discarded part of the "pure" information so that higher compression ratio can be achieved, as shown in Figure 1.2. In that case, since some of the entropy is lost, the original data would not be correctly reconstructed after decoding. Therefore, this type of compression is named **Lossy** Coding. This thesis mainly focuses on lossy video coding.

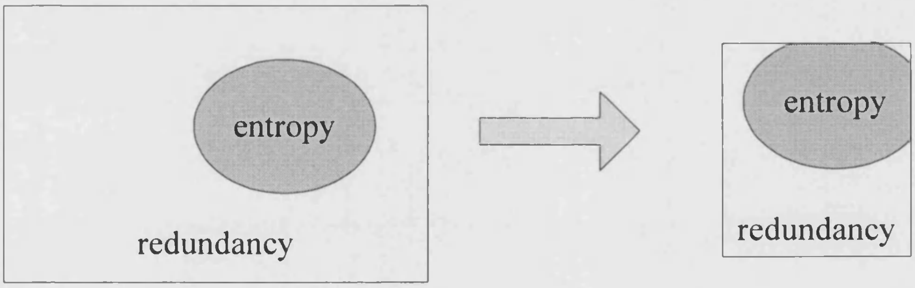


Figure 1.2 Lossy Data Compression

Lossy compression introduces error to resource data, which means quality degradation in video coding. However, it is still popular in video applications, where a higher compression ratio is required and more distortion is acceptable. Such approaches can be applied to video is because of the psycho-visual redundancy. Although psycho-visual redundancy is part of the entropy, removing it can be carried out without causing any perceivable affect to the video quality. This type of redundancy mainly lies in two ways. Firstly, the temporal and spatial resolution might be higher than the human eyes can perceive. Examples of exploiting such redundancy include sub-sampling frames and lower spatial resolution of chrominance components. Secondly, some part of the images might be complex and expensive to code. But, if this part is not sent, the perceived error caused is rather small.

1.1.2.2 The role of rate control

In image and video compression, the compression ratio is defined as:

$$Compression\_ratio = \frac{Original\_Size}{Compresed\_Size} \qquad \textbf{(1.2)}$$

Compression ratio is used to measure how much compression is achieved. Theoretically, the compression ratio can be always increased by discarding more and more information, which would cause more and more deviation of the decoded data from the original.

The resultant size of video signal after compression is referred to as *bit rate*, often denoted as *R*. For image, this is often measured by *bits per pixel*, while for video by *bits per second.* The difference between the original and compressed video sequences, or error, is often called *distortion*, denoted by *D*, which is normally measured with values such as mean squared error (MSE). More details of MSE are given in Chapter 2.

Lossy compression not only reduces signal size, but also reduces the quality. The price for lossy coding to achieve a higher compression ratio is to introduce more error, or distortion to the data. For any lossy compression application, it is always necessary to consider the trade off between the rate and the distortion. A higher compression ratio, or a smaller compressed data size, results in higher distortion, namely lower quality, and vice versa. Preserving higher quality requires higher rate to be used. The video quality must be maintained at an acceptable level.

As lossy coding can achieve much higher compression ratios, it is more widely used and rate distortion control is of essential importance in applying lossy compression. The relationship between rate and distortion can be formulated by a function $D = f(R)$, where $f()$ is the rate distortion function. The objective of Rate Distortion control is to achieve the best possible quality at given bite rate, which can be stated as,

$$\textbf{Minimize } \boldsymbol{D}, \quad \textbf{subject to: } R \le R_{\max}\,. \qquad \textbf{(1.3)}$$

The optimal solution to the above constrained optimization problem, for any given rate R_{max}, results in the operational rate distortion curve of a video coding scheme. This thesis mainly discusses how to achieve this in digital video coding.

1.2 Relevant work

It is common in visual compression to apply a three-stage approach, which consists transformation, quantization and entropy coding. And here quantization is the part that is responsible for deciding how much information to discard. Rate control is mainly about how to manipulate the quantization step.

1.2.1 Typical rate control schemes

To achieve further compression, a video coder often exploits the motion prediction and compensation. This introduces motion vectors and classifies the frames into different types. All these need to be considered in video rate control, which makes it more complicated.

1.2.1.1 MPEG-2 Rate control algorithm

TM5 [2] is a well known test version of the MPEG-2 [3] reference code. MPEG-2 uses a group of pictures (GOP) structure, which consists of at least one I frame and a few P and B frames. The TM5 rate control algorithm has two steps. It first uses a bit allocation scheme to set the target bits for each frame within a GOP. Then it carries out a macroblock level rate control basing on the buffer status and macroblock's spatial activity. This whole scheme is based on some assumptions:

- The distortion D decreases linearly with the quantization parameter *q*.
- Because of the characteristic difference among I, P and B frames, the quantization parameter q should be set differently for each frame type. Denoting the parameter for each frame type as q_I , q_P , and, q_B a relationship between them can be written as:

$$\frac{q_I}{1.0} = \frac{q_P}{c_P} = \frac{q_B}{c_B} \qquad \textbf{(1.4)}$$

By default, c_P and c_B are set to be 1.0 and 1.4.

- The coding rate R is inversely proportional to the distortion D, and thus we can have:

$$R = \frac{const}{D} \tag{1.5}$$

It is obvious that this rate control algorithm employs a few assumptions that simplify the Rate-Distortion relationship. It however cannot achieve a robust and accurate rate control.

1.2.1.2 VM8 Rate control algorithm

In verifying Model 8, or VM8 [4], a rate control scheme is designed for MPEG-4 [5]. This scheme makes use of the knowledge obtained from coding previous frames to estimate the R-D behaviour of the current frame. Normally the neighbouring frames are very similar in content and so is it in R-D relationships. The following assumptions are made in this scheme:

- Adjacent video frames of the same type have the same rate and distortion curves.
- The curve of the relationship R(q) of each frame is simulated by a formula,

$$R(q) = a_1 \times q^{-1} + a_2 \times q^{-2} \tag{1.6}$$

  After a frame is coded, the average quantization parameters and the total bit rate are recorded. Such statistics over a number of frames are then used to estimate the model parameters a_1 and a_2 for the current frame. Then the above model can be used for rate control.

VM8 suffers from possible scene changes, which means the R-D relationship might change drastically. Besides, because of the limited accuracy of the mathematical model the control error might be large under some circumstances.

1.2.1.3 TMN8 Rate control algorithm

TMN8 [6] is designed for H.263[7]. This rate control algorithm operates at macroblock level. It accumulates the coding statistics of previous macroblocks to update the parameters in the R-Q model below,

$$R(q)=\begin{cases}\frac{1}{2}\log_2(2e^2\frac{\sigma^2}{q^2}) & \textbf{if} \quad \frac{\sigma^2}{q^2}>\frac{1}{2e} \\ \frac{e}{\ln 2}\bullet\frac{\sigma^2}{q^2} & \quad\quad \frac{\sigma^2}{q^2}\le\frac{1}{2e}\end{cases}, \qquad (1.7)$$

Compared to the VM8 rate control scheme, this model is more complex and much more accurate. It helps to meet the target bit rate precisely, and to maintain a steadier buffer level. Yet, this rate control scheme does not regulate the dynamic range of quantization parameters of each macroblock. The model also suffers from low accuracy problems at scene changes. Also it is designed for P frames and I frames use a much rougher model and thus does not work well, which means difficulty in practice due to the necessity of frequent refreshing of intra coding.

1.2.2 Other relevant work

The problem of rate control has been studied intensively, and there are works studying aspects other than those discussed above of the problem. The bit allocation between motion vector, mode, and residual is studied in [8-10]. The work in [11] is an extension of these schemes, and the technique in [11] is also applied in H.264 [12, 13].

Reed and Lim [14] proposed an optimal coding scheme that jointly optimizes the temporal and spatial quality. It simultaneously adjusts frame rate and quantization parameters using dynamic programming. This algorithm assumes that all frames are coded as intra images. The I frame restriction, which is adopted to avoid the computational difficulty in dealing with prediction dependency, imposes a severe constraint on its applicability to motion compensated video coding schemes, such as ISO/IEC MPEG [3, 5, 15] and ITU-T H.26X [3, 7, 12, 16, 17] codec families.

The application range was extended by Liu and Kuo [18], to a wider range of cases. This work investigates the joint temporal-spatial bit allocation with dependency for video consisting of both intra- and inter-coded frames. All frames are considered as a stage of the coding process. And for each frame, every possible quantization parameter is treated as a node. The coding paths connecting all the nodes at various stages build up a "trellis". Exploiting the Lagrange multiplier method, in which the coding cost is defined as a value combining the coding rate and the distortion, the scheme tries to truncate inefficient paths using the cost as a standard. "Skip frame" mode is included as a node in each stage so that the frame rate could be modified as well.

The two schemes discussed briefly in this section look deep into the problem of optimal bit allocation for video. The optimization process is very calculation intensive and thus both algorithms involve very high complexity. Generally speaking, an exhaustive search is required to guarantee the optimal solution by examining all possible coding options.

From the introduction in section 1.2.1, it is seen that it is important to have a precise source model to simulate and predict the bit and distortion change. A lot of work has been done to develop such models. Ribas-Corbera and Lei [19] proposed a quadric model to predict the distortion from the quantization step size. The model was proposed for .263+, and is widely used in many following schemes [20].

He and Mitra [21] proposed a linear ρ domain mathematic model for rate estimation. It turned out to be very precise and becomes a very powerful tool in rate control. This algorithm develops a function describing the relationship between the percentage of the zero valued coefficients after quantization, and the output bit stream size. Experiments show that there is a near-linear relationship between those two values. This simple model shows very high accuracy which enables very robust and precise rate control. Several works [22-24] have been done based on this model. However, in allocating the final quantization parameter it has to choose from limited options. Inevitable deviation still exists even if the quantization parameter that leads to the output bits number that is closest to the target value is chosen. Also, applying this scheme to H.264 would cause difficulty in counting zeroed coefficients due to the special quantization schemes.

1.3 Objective

This objective of the thesis is to improve the video coding efficiency by rate control optimization techniques. It studies the latest video standard and the application of matching pursuit in video coding in order that advantages of various techniques can be combined together to achieve improved coding efficiency.

1.3.1 Improved conventional hybrid video coding with operational rate distortion optimization

The H.264 [25] is the latest video coding standard and provides the highest coding efficiency in literature [26, 27]. To achieve high compression efficiency, the rate distortion optimization (RDO) scheme is introduced in H.264. This helps to choose the best coding modes during encoding, but also makes it hard to perform rate estimation and control. The model [28] exploited in rate control needs to calculate the quantization parameter (QP) from the distortion to allocate the target rate, while the QP must be first decided before the modes and distortion can be decided. This is referred as chick and egg dilemma[29]. Existing schemes [29, 30] try to predict the QP value and apply a two pass scheme to solve this problem. This obviously lacks accuracy. An algorithm able to avoid such problem would potentially improve the ultimate coding efficiency.

1.3.2 Applying scalable coding scheme to video coder

Conventional macroblock (MB) based image compression has difficulty to develop a straight relationship between the quantization step and the bits number. Embedded image coders [31-33], in contrast, allows a highly scalable bit stream so that the target bit rate can be easily achieved with high precision, and with no need of repetition.

Adopting such a scalable frame coder into a video system would help to accurately achieve the target rate in video coding.

1.3.3 Low complexity rate distortion source modelling

Accurate source modelling is of great importance for high coding efficiency. The rate models in literature normally study the relationship between the rate and some statistic values, such as data variations. To improve the accuracy, the models tend to be very complicated. Since we are applying different frame coder, new work must be done to reveal the source character.

1.3.4 Bit allocation control based on the R-D model

With the help of the source model for the new video coder, as described above, bit allocation can be optimized to improve the compression performance. Optimization techniques justify the bit usage for various part of the source information so that the distortion could be minimized.

1.4 Approach

For the objectives presented in section 1.3, we apply the following methodologies.

1.4.1 Operational rate distortion optimization for transform coding

The RDO scheme in H.264 introduces efficiency improvement as well as difficulty in rate control. Considering that the QP value is normally required not to change too much in adjacent coding blocks [34], there are only very limited number of candidate QP values. Applying Lagrange multiplier scheme to pick from the candidate values could avoid the difficulty from the dilemma, and also provide improved performance. This would be usable for non-real time applications.

1.4.2 Applying matching pursuit to video coding system

The Bath University Matching Pursuit (BUMP) project designs new matching pursuit algorithm that not only provides outstanding coding efficiency, but also produces highly scalable bit stream. Adopting such scheme into a video system that combines the advantage of both BUMP and the motion model in H.264 constructs a new system that might give better features. A simple bit rate tuning scheme exploiting the scalability of the bit stream is also adopted.

1.4.3 Rate and distortion source modelling for matching pursuit algorithm

The new BUMP algorithm has its own character. The specific coding mechanism of MP indicates a simple relationship between the atom amplitude and the resultant distortion. And the distinctive quantization scheme of BUMP limits the number of possible reconstructed values. This means the distortion of the frame after BUMP coding can be predicted in a simple way. Studies are also carried on over the relationship over the bits number and the atom number. A simple mathematic model is developed for bits modelling, and an updating scheme is developed so that the bits number can be predicted with very high precision.

1.4.4 Rate and distortion optimization for matching pursuit video coding

A rate distortion slope (R-D slope) unifying scheme is derived from the Lagrange multiplier algorithm. The unique BUMP scheme and the R-D source model enable such slope calculation and unifying process to be carried out. The BUMP allows the R-D slope of coded atom to be calculated, and the model enables the future slope to be predicted. Decision of the best truncation point can be made by comparing the slopes and the coding performance can be improved.

1.5 Thesis outline

The rest of the thesis is organized as follows,

Chapter 2 reviews the relevant background knowledge. It first presents the basic 3-block visual coding approach, and then each stage of the approach is briefly reviewed. It then reviews how the rate and distortion is measured, and how optimization techniques are performed. After the background review, a unified video coding framework is proposed. The framework is converted from existing video reference software and provides a variety of functionalities that are necessary for research work. The experiments and discussions in later part of the thesis are all carried out on this platform.

Chapter 3 presents an operational rate distortion optimization scheme for H.264. It first studies the special quantization implementation and RDO technique in H.264, and then an R-D slope unifying approach of optimization is proposed. Further experiments show that this approach has its limit and is not appropriate to be applied at macroblock level. Finally, the Lagrange multiplier is applied as the optimization approach. By properly choosing the candidate QP values, the repetition of encoding is reduced and thus the complexity is reduced. The problem is converted into a path picking problem. Details of the performance are then discussed.

As another type of image coding schemes, embedded coding provides an easy approach for rate control problem. Chapter 4 proposes the idea of applying such techniques into video system. Two typical embedded techniques in literature are chosen to be adopted into the video framework. Experiments show that these techniques do not afford as high efficiency as H.264 system does. For a better embedded coding scheme, matching pursuit is proposed as that candidate embedded frame coder. A simple and effective bit rate tuning method is then designed which effectively improves the final compressed video quality.

In Chapter 5, the specific features of BUMP are studied to develop a model that depicts the source R-D character. The tuning method in Chapter 4 is ad hoc, and for more general applications and better optimization, such a model is required. The coding mechanism of MP enables easy distortion measure, and the difficulty left is rate estimation. The PLQ and MERGE parts in BUMP are studied in depth, and finally, a simple and highly precise model for rate estimation is then proposed. A highly precise adaptive rate estimation method is proposed based on that model.

The R-D optimization for BUMP coder is studied in Chapter 6. The slope unifying method proposed in chapter 3 is difficult to be applied on macroblock basis. However, in the case of encoding a frame with MP, the slope can be calculated with a single pass using the source model, and the optimization of unifying can be easily performed. With the model developed in Chapter 5, this R-D optimization scheme is possible. The scheme is applied to the hybrid BUMP video coding scheme, and comparison between the proposed scheme and the original H.264 rate control scheme are discussed.

The conclusion of the thesis is summarized in Chapter 7, and some future work directions are also briefly discussed.

Chapter 2 BACKGROUND

The problem of video compression involves a lot of knowledge and techniques of various aspects and subjects, such as signal processing, transformation, information theory, and optimization methods. Following presents some key background knowledge for video compression.

2.1 Analogue and digital video

A typical natural scene is composed of various objects, including background, each having its own characteristics, such as shape, colour, texture, and illumination. A traditional optical camera generally consists of a lightproof enclosure having an aperture with a shuttered lens through which the image of an object is focused and recorded on a photosensitive film.

Similarly, a natural video clip is composed of “real world” objects, involving continuous movements. Analogue video signals are normally generated at the output of a camera by scanning a two-dimensional moving scene and converting it into a one-dimensional signal [35]. A moving scene is a collection of individual images, where each scanned picture generates a frame of picture. Scanning starts at the top left corner of the picture and ends at the bottom right. There is a lower limit to the number of pictures per second, below which flicker becomes perceptible.

Digital video signals can be obtained by digitizing analogue video signals. This normally involves the three steps of filtering, sampling, and quantization. The filtering operation is employed to avoid the aliasing artefacts of the following sampling process.

Sampling is performed upon the filtered signals to generate discrete time signal. For a video signal, the sampling is done in both temporal and spatial domains. Similar to analogue video, the temporal domain sampling results in a series of frames. Sampling in the spatial domain converts the frames into digital images composed of individual pixel values.

Temporal domain sampling captures the movements in successive frames at regular periodic intervals. The higher sampling rate, or lower periodic intervals, provides a perceptually smoother motion in the video, and requires more frames to be recorded and stored. In practice, the frame rate is decided by the specific need. Frame rates of 25 or 30 frame per second (fps) are commonly found in the standards of television video signals. Frame rates between 10-20 fps can be used for low bit-rate communication. Frame rates below 10 fps are sometimes used for very low bit rate video communication, where motion can be very jerky and rough, since a comparatively large amount of data is discarded. In contrast, frame rates like 50, 60 fps enable very smooth motion quality at the cost of much larger data rate, and are found in high resolution video services [35].

A sampled frame is a rectangle made of evenly positioned sampling points, called pixels. Comparatively, the spatial sampling rate defines how many pixels are used to represent each frame. A frame from digital video can be regarded as a two dimensional function of *f(x, y)*, where *x* and *y* are spatial coordinates, and the amplitude of *f* at any pair of coordinates *(x, y)* is called the intensity or grey level of the image at that point. Spatial sampling makes this function discrete in the *x* and *y* coordinates. Obviously, the visual quality is affected by the number of pixels. Higher sampling rates provide higher frame resolution, enabling smoother continuous-tone imaging. A coarse sampling grid produces a low resolution image, in which the image may look blotchy.

Quantization is the final step of digitizing. It is common in all digital signals that the values are represented in binary form with limited number of digits. For digital video signals, they are normally quantized into eight-bit resolution, which is suitable for video broadcasting application. After digitization, a video signal is converted into a series of digital images which are saved as matrices of discrete pixel values. Figure 2.1 shows the structure of pixel, frame and sequence in a digital video.

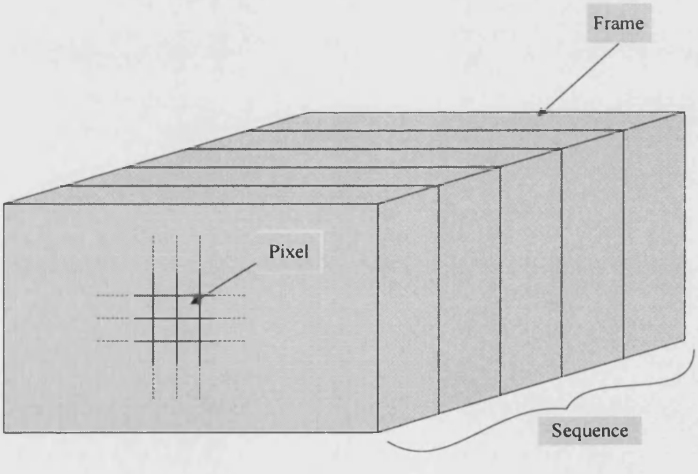


Figure 2.1 Pixel, frame, and sequence

2.2 Video compression

It is introduced that digital image and video signals contain huge amount of data, which makes it hard to transmit and store them. Compression is then necessary to make many applications of digital video possible in practice.

2.2.1 Redundancy and compression

It has been introduced in section 1.1.2 that Shannon's theory [1] gives a measure of entropy, and the data can be divided into entropy and redundancy.

Video compression is possible mainly because of the redundancies existing in video signals. Removing those redundancies can reduce the data size while still reserve the information that is desired by the users. The following are the main types of redundancy to be exploited in video compression:

1. Spatial redundancy: A video sequence is composed of a series of digital images. A statistic analysis indicates that there is normally a very strong correlation within picture content of each frame. This is seen as similarities of the pixel values within a local area. De-correlating the data can help to achieve data compression.

2. Temporal redundancy: the temporal sampling nature of video means there are strong similarities between successive frames. By coding the differences between frames, the video size can be reduced radically.

3. Coding redundancy: It is suggested in [1] that optimal number of bits for coding a symbol depends on the probability of its appearance. The variable length coding can be employed to reduce the redundancy among the compressed symbols.

4. Psycho-visual redundancy: some part of the information might be lost and no perceivable quality degradation can be caused. For example, losing some details in colour representatives is much more tolerable to human eyes than in luminance components [35, 36].

2.2.2 The basic building blocks in digital image coding

According to the above types of redundancies, many techniques are designed to remove them. In this thesis, when the term of "coding" is used for visual signals, it refers to the same meaning of "compression".

2.2.2.1 Predictive image coding

From the definition of spatial redundancy, it is a direct approach to de-correlate the pixels by predicting a pixel from its neighbouring pixels' values and then coding a prediction error rather than the original value [37]. Predictive coding can be used for both lossy and lossless coding schemes.

The lossless coding mode of the JPEG standard is a typical predictive coding scheme[38], as depicted in Figure 2.2. In this predictive coder, an input pixel *x* is predicted by weighted combination of its three neighbouring pixels' values the *a*, *b* and *c* as in Figure 2.2. In the practical coding, the encoder needs to decide the best choice of combination to use, and send the decision through the bit stream, so that the decoder will perform the correct operation to decode the image.

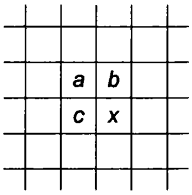


Figure 2.2 Three-sample prediction neighbourhood

Such prediction effectively removes the significant spatial correlation existing in images before the actual coding is carried out and the prediction errors it produces are much smaller in amplitude and contain much less variation. Obviously, coding the errors rather than the original image allows easier coding and the information can be conveyed at a greatly reduced bit rate. Thus, compression is achieved.

2.2.2.2 Transformation for image coding

A more widely used approach for removing spatial redundancies in images is to exploit transformation. It works by mapping the pixels into a transform domain prior to data reduction. The pixels of most natural scenes are highly correlated, and the image energy is mainly concentrated in the low frequency region. Therefore, in the transformed domain the energy is mainly carried only by a few coefficients. Transformation de-correlates the pixels values and makes the image energy compact. Simple coding may be more effective in such a domain than in the original visual signal space.

Transform coding has been widely studied and has been very popular in the past two decades. The most widely chosen transformations are discrete cosine transformation (DCT) [35, 39] and discrete wavelet transformation (DWT) [40, 41]. These two transforms are adopted in most digital video coding standards [38, 42-52].

The definition of DCT [39] is:

Forward transform $$F(u)=\frac{1}{\sqrt{N}}C(u)\sum_{x=0}^{N-1}f(x)\cos\left(\frac{\pi(2x+1)u}{2N}\right)$$ u=0,1,.......,N-1 **(2.1)**

Where $$\begin{cases} C(u)=\frac{1}{\sqrt{2}},u=0; \\ C(u)=1,u\neq 0; \end{cases}$$

Inverse transform $$f(x)=\frac{1}{\sqrt{N}}C(u)\sum_{u=0}^{N-1}F(u)\cos\left(\frac{\pi(2x+1)u}{2N}\right)$$ **x= 0,1,....,N-1 (2.2)**

The above equations give a one dimensional DCT. Since a natural image is two dimensional data and correlations among the pixels exist in both horizontal and vertical directions, a two dimensional DCT is required. A two dimensional DCT is a separable process that is achieved using two one-dimensional DCT, one for horizontal and the other for vertical direction.

A two dimensional DCT can be defined as,

$$F(u,v)=\sqrt{\frac{2}{M}}C(v)\sum_{u=0}^{N-1}F(u,y)\cos\left(\frac{\pi(2y+1)v}{2M}\right),$$ **v = 0,1,....M-1** **(2.3)**

Figure 2.3 shows the resultant basis functions when M and N are 8.

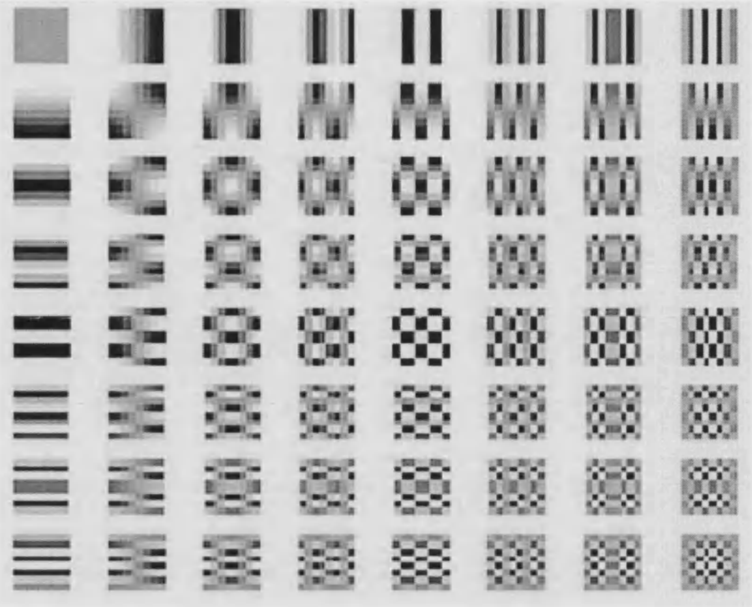


Figure 2.3 Basis functions of 8x8 DCT

Theoretically, DCT of larger size exploits spatial correlation within a larger area, and therefore enables higher coding efficiency. However, the calculation complexity of DCT tremendously increases when the size increases. Thus the practical transform size is a compromise between coding efficiency and complexity. In JPEG [38, 51] and many video standards [2, 43, 44, 52], DCT transformation of size 8x8 is exploited.

Another popular transformation method in image compression is discrete wavelet transformation (DWT), and many coding algorithms have been developed based on it. [31-33, 53]. The latest image compression standard JPEG2000 [49, 54, 55] is also based on DWT.

Wavelet transformation was first introduced as a signal analysis tool [40, 41]. In the past two decades, it has attracted wide range of interest and showed extremely good performance in image compression [31-33, 53, 54, 56, 57]. The principal idea of the wavelet transform is to represent any arbitrary function *f* as a superposition of wavelets. It breaks the signal into its "wavelets", scaled and shifted versions of the "mother wavelet". Once the mother-wavelet $\psi(t)$ is defined, the family of wavelets can be obtained from the definition:

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-a}{b}) \qquad (2.4)$$

The members of the wavelet family can be used to decompose a square-integrable function *x(t)* into a set of basis functions, such as

$$X_{\omega}(a,b) = \int_{-\infty}^{\infty} x(t)\psi_{a,b}(t)dt \qquad \textbf{(2.5)}$$

It can be seen that a one dimensional signal *x(t)* is mapped into a two dimensional function $X_{\omega}(a,b)$ and this makes it very redundant. The original signal can actually be recovered from the wavelet transform obtained from discrete values of *a* and *b* [58]. In practice, *a* can be made discrete by choosing $a = a_0^m$, with $a_0 > 1$ and *m* an integer. Similarly, *b* can be made discrete which corresponds to sampling in time. The sampling frequency depends on the bandwidth of the signal and should be proportional to *a*. So it can be chosen as $b = nb_0a_0^m$. In practice, $a_0 = 2, b_0 = 1$ are often chosen. The original signal can then be represented as follows:

$$x(t) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \alpha_{m,n}\psi_{m,n}(t) \qquad \textbf{(2.6)}$$

where

$$\alpha_{m,n} = \int_{-\infty}^{\infty} x(t)\psi_{m,n}(t)dt \qquad \textbf{(2.7)}$$

The most widely used method to implement DWT is filter banks [41] as shown in Figure 2.4.

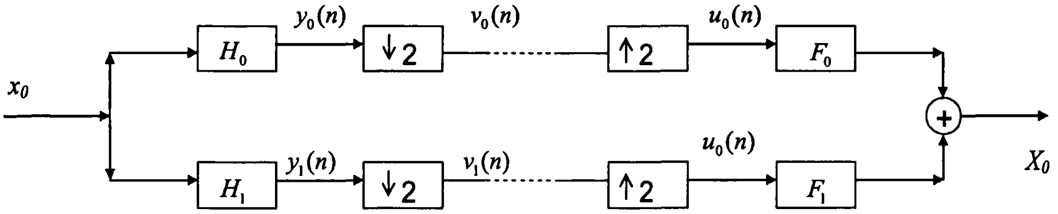


Figure 2.4 Filter bank for wavelet

A basic wavelet filter bank structure consists of an analysis bank, and a synthetic bank. In Figure 2.4, the analysis bank is composed of H_0 and H_1, and the synthetic bank is composed of F_0 and F_1.

In a filter bank, a signal is decomposed into two channels at the analysis bank, where the low-pass filter (H_0) averages the signal, and the high-pass filter (H_1), which lets through the details of the signal. The coefficients from both channels are down-sampled, and then fed into the synthesis bank for reconstruction. It is common to apply DWT to the low frequency coefficients repetitively to achieve a better energy compaction. Figure 2.5 presents an example of result of an image after 3-scale DWT transformed.

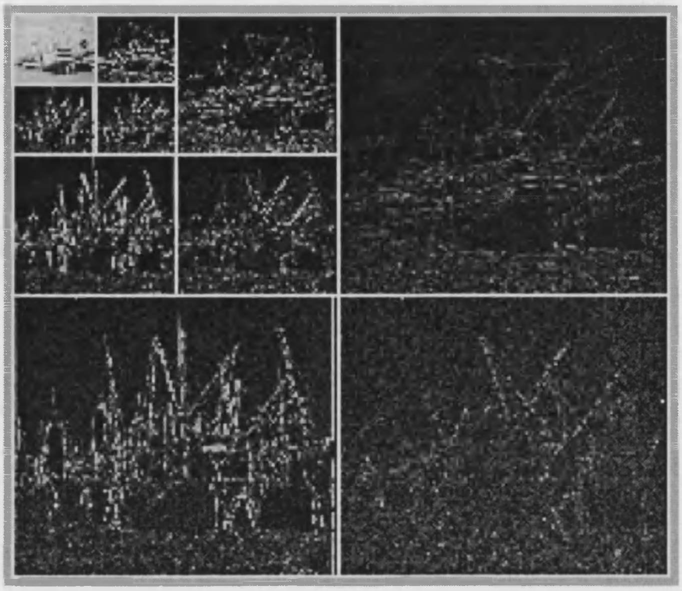


Figure 2.5 A wavelet transformed image

Unlike the Fourier Transform, which is unable to associate features in the frequency domain with their location in time, wavelet provides a combination of time (space) and frequency resolutions. That is, from the transformed coefficients, we know both the frequency components and the time (spatial) location of these components. This assures a very good basis for image coding.

Antonini, Barlaud, Mathieu, and Daubechies [53], first used discrete wavelet transform for image compression, and attracted some attention. Later Shapiro [31] designed the famous EZW image coder, which intensely surpasses JPEG in image quality. This heralded the success of wavelet in image coding. Later, many more wavelet based image coders [32, 33, 56, 57, 59] have been proposed.

2.2.2.3 Quantization

Quantization prepares the transformed coefficients so that they can be more effectively entropy coded in the next stage. Quantization is different comparing to other parts in that information loss happens in this stage, and this is why the video coder is called lossy.

Implementation-wise, quantizing a number can be summarized as this: let O, Q and ϑdenote the original number, the quantization factor and the quantized number respectively, the quantization process is then defined as,

$$O_1 = \left[\frac{O}{Q}\right] \tag{ 2.8}$$

and the de-quantization entails the following operation:

$$O' = O_1 \times Q \tag{2.9}$$

where $O'$is the reconstructed number.

Quantization and de-quantization are basic division and multiplication. But, in practice, ϑ is often kept as integer value only, and after de-quantization, only an approximation of the O can be reconstructed. In equation (2.12), we use $[\bullet]$to represent the process of discarding the real part of the quotient. In programming, this process can be carried out in various ways, including functions such as *fix()*, *round()*, etc.

Through quantization, the numerical precision of O is effectively reduced, for example, if a particular number has a range of [0, 999], there are 1000 possible numbers. But if the number is quantized (divided) by 10, the quantized numbers will only range within [0, 99] for a total of 100 possible numbers.

It is worth mentioning that the quantized values are normally rearranged and mapped into an even smaller set of symbols. For example, that JPEG [38, 51] exploits the DCT and then maps the AC (high frequency) coefficients into two dimensional (RUN, CAT) events. Such reduced size of number set makes it much easier for entropy coding later on.

Quantization causes information loss, and hence video quality degradation, and reduced data size. Therefore, this is very important for rate and distortion control. This will be further discussed this with more details in later sections.

2.2.2.4 Entropy coding: removing the coding redundancy

Shannon's theory [1] gives the minimum average bits necessary for representing the symbols. It is calculated as

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{2.10}$$

Any extra coding bits on top of that is called *coding redundancy.*

Entropy encoding is a scheme that assigns codes to symbols so as to match code lengths with the probabilities of the symbols. Entropy coding is responsible for removing coding redundancy and thus makes the code stream more efficient. The quantized transformation coefficients are then to be entropy coded to reduce the data size. Entropy coding is a lossless process that reserves all the information involved. Two commonly found entropy coding schemes are Huffman coding and arithmetic coding.

Huffman coding might be the most widely known variable length coding method and was developed by David A. Huffman [60]. Huffman coding assigns variable length output code to each symbol according to the symbol's probability. The fundamental idea is to allocate short code word to a symbol with higher probability and longer code word to that with smaller probability.

The process of allocating code word takes the following steps:

- Rank all the symbols in the order of their probability of occurrence.
- Successively merge every two symbols with the least probability to form a new composite symbol, and re-rank it with all the other symbols before merging, each time allocate 0 and 1 to the two symbols respectively. This will create a tree structure, where each node is the probability of all nodes beneath it.
- Trace a path to each leaf, and collect all the bits along the path to make up the code word for that symbol on the leaf.

Figure 2.6 shows an example of Huffman coding.

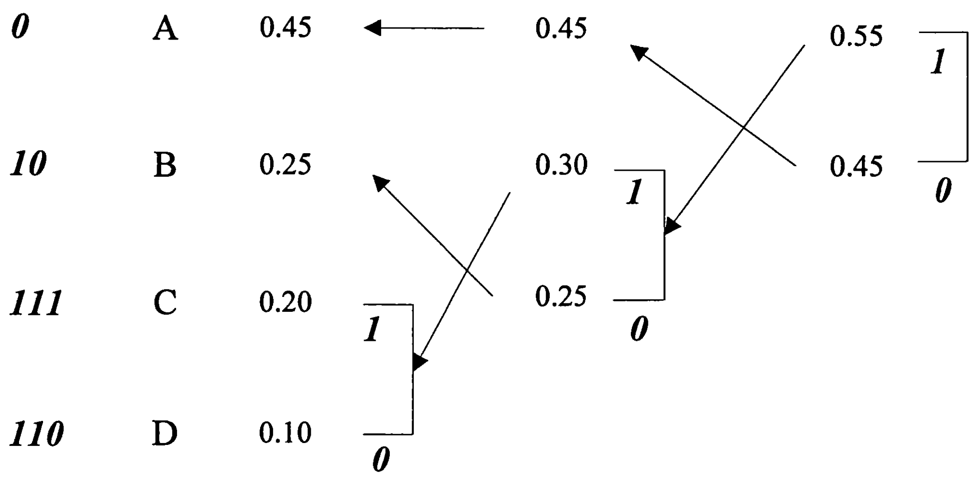


Figure 2.6 Example of Huffman coding

The optimal number of bits for a codeword given to a symbol with the occurrence probability of *p* is $-\log_2 p$, which is not an integer unless *p* is an integer power of 1/2. However, the Huffman coding scheme only allocates code words of integer number of bits. If a symbol's entropy is not an integer power of 1/2, on average this will lead to more than the symbols' entropy value of bits to code the symbol. This makes Huffman coding suboptimal in many cases.

Arithmetic coding is a data compression technique that is capable of effectively coding symbols whose entropies are not exact integers. It encodes data by creating a code string, which represents a fractional value on the number line between 0 and 1 [61].

Arithmetic coding allocates an interval of real numbers between 0 and 1 to a set of symbols. The length of the interval depends on the symbol's occurrence probability. As the message becomes longer, the interval becomes shorter, and more bits are needed to specify the interval. Since the model allocates larger ranges to the symbols of higher probability, they would then reduce the interval less, and therefore contribute fewer bits to the message.

The two tables in the following show an example of arithmetic coding.

Table 2.1 shows a fixed model of range values for a set of alphabets and Table 2.2 shows the range change for a message.

Symbol	Probability	Range
A	0.2	[0.00000,0.20000)
B	0.3	[0.20000,0.50000)
C	0.1	[0.50000,0.60000)
D	0.2	[0.60000,0.80000)
E	0.1	[0.80000,0.90000)
F	0.1	[0.90000,1.00000)

Table 2.1 Allocation of value intervals to symbols

	New character	Range
Initially		[0.00000,1.00000)
After seeing a symbol	B	[0.20000,0.50000)
	A	[0.20000,0.26000)
	C	[0.23000,0.236000)
	C	[0.23300,0.23360)
	F	[0.23354,0.23360)

Table 2.2 An example of coding a message with arithmetic coding

Arithmetic coding does not allocate a specific code word to a symbol but separates the model for representing the data and encoding information with respect to the model.

Therefore, compared to Huffman coding, arithmetic coding avoids the restriction that a symbol has to be interpreted as an integer number of bits. It then achieves a bit stream very close to the theoretical entropy bound.

Both the above two entropy coding schemes require a probability model. Such models can be fixed, or adaptive along with the coding process. The choice between them should be made according to the practical applications.

2.2.2.5 The three-block structure of image coder

Generically, a transform image coder consists the three stages introduced earlier this section-transformation, quantization, and entropy coding, as shown in Figure 2.7. It is characterized mainly by its transformation part, which is carried out before any further coding process.

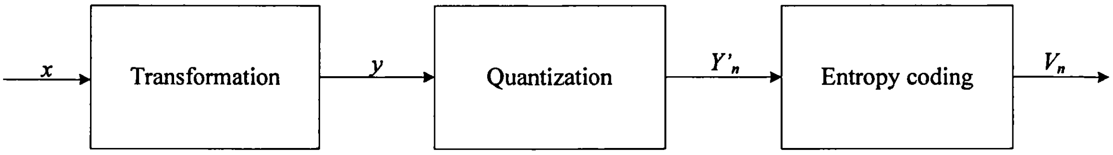


Figure 2.7 Three stage model for transform coding

Through transformation, the spatial domain pixels are mapped into transformed domain coefficients. Theoretically, transformation reserves all the information from an original image, that is, the original image can be precisely recovered by performing the inverse transformation over the coefficients. However, transformation de-correlates the pixels, and makes the energy compact in the transformed domain. Quantization and entropy coding, which are performed in succession, are much easier and more effective to be performed on such de-correlated coefficients than on original pixels. The 3-stage structure is also widely applied in video coders, as can be seen in following sections.

2.2.3 Hybrid video coding schemes

2.2.3.1 Motion JPEG: JPEG for coding video

It was addressed in section 2.1 that digital video is a succession of still images. Therefore, a video sequence can be coded in such a way that each single frame is encoded by JPEG. In this case the process is called motion JPEG.

Motion JPEG has found numerous applications. It is widely used in non-linear editing systems that require accessing any frame in a video clip with the same ease as any other [62]. This is possible with motion JPEG since all frames are coded independently. For a similar reason, motion JPEG is also utilized for transmission over internet where congestion might last for a long time because of unpredicted bandwidth and network load. Another advantage of motion JPEG is the resilience to information loss because the error cannot propagate through the sequence of independent images [35]. However, the correlation among the successive frames is not exploited, and the redundancy in the temporal direction is not considered by motion JPEG. To achieve higher compression ratio, the temporal redundancy must be removed.

2.2.3.2 Motion estimation and compensation

Motion estimation is introduced to be responsible for removing the temporal redundancy and can be carried out in a few different fashions.

In a typical block matching algorithm, a frame is divided into *M*x*N* pixel blocks. More often, these blocks are *N*x*N* squares. Then, for a maximum motion displacement of *w* pixels per frame, the current block of pixels is matched against a corresponding block at the same coordinates' size but in the previous frame, within the square window of width *N+2w*, see Figure 2.8. The ultimate displacement is obtained from the best match according to a matching criterion.

Various measures can be used as the matching criterion such as the cross correlation function (CCF), mean square error (MSE), and mean absolute error (MAE). For best matching, the CCF needs to be maximized, while if the latter two are applied, the errors are to be minimized. In practical coders, the last two are used since CCF is believed to be not effective in tracking motions [35, 63]. The matching function of MSE is defined as:

$$M(i,j)=\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}(f(m,n)-g(m+i,n+j))^2, \qquad -w\leq i,j\leq w \qquad \textbf{(2.11)}$$

and for MAE,

$$M(i,j)=\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\left|f(m,n)-g(m+i,n+j)\right|, \qquad -w\leq i,j\leq w \qquad \textbf{(2.12)}$$

where *f(m, n)* represents the current block of N2 pixels at coordinates *(m,n)* and *g(m+i,n+j)* represents the corresponding block in the previous frame at new coordinates *(m+i, n+j)*. At the best matched position of *i=a* and *j=b*, the motion vector, MV (*a*,*b*) represents the displacement of all the pixels within the block. This process is illustrated in Figure 2.8.

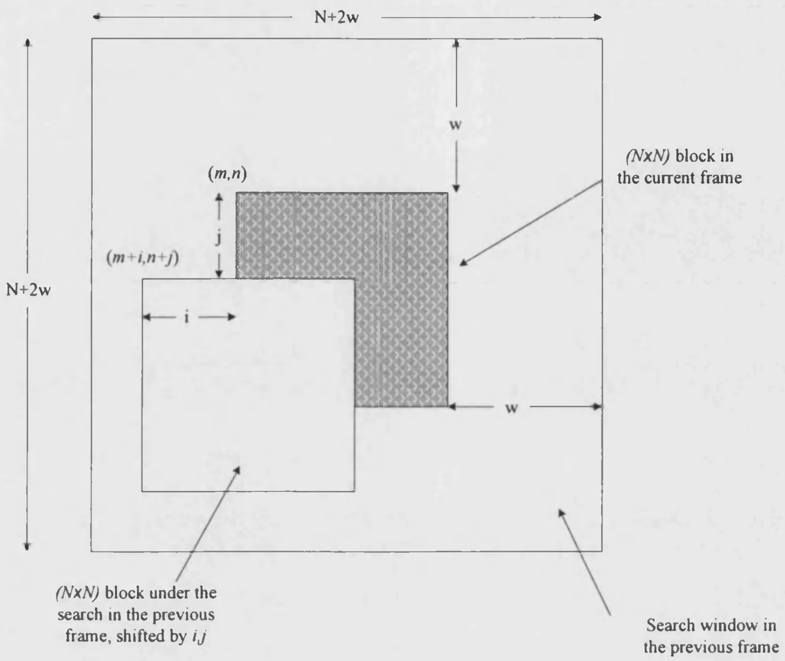

N+2w
N+2w
w
(*NxN*) block in the current frame
(*m*,*n*)
j
(*m+i*,*n+j*)
i
w
(*NxN*) block under the search in the previous frame, shifted by *i*,*j*
Search window in the previous frame

Figure 2.8 Block-based motion estimation

Many techniques can be used for motion estimation such as fast motion estimation, hierarchical motion estimation and so on. Some also consider motion bit rate and distortion together for better optimization [9, 11].

Motion prediction can be carried out forward to predict future pictures, or backwards to predict earlier pictures. According to the motion prediction carried out, frames in a video sequence can be divided into I (Intra) frames, P (Predicted) frames and B (Bidirectional) frames. An I frame is coded only by exploiting the correlation within the image and no motion prediction is involved; a P frame is predicted from a previously coded I frame or P frame, and a P frame is also used to predict other frames; a B frame is said to be bidirectional as it is predicted from both forward and backward directions. P and B frames are also called Inter frames. In video coding, it is often required that I frames should be inserted on a regular basis to prevent any possible errors from propagating among the P frames. B frames are not used to predict other frames and so can be used without the problem of error propagation.

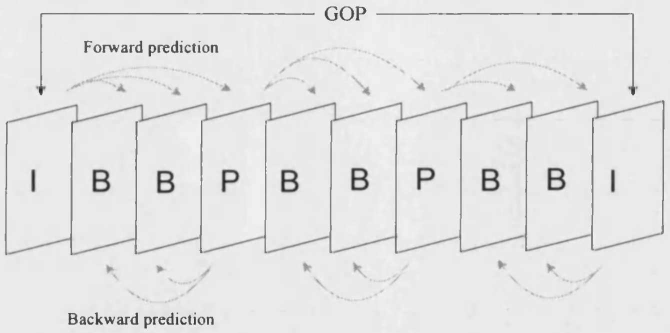


Figure 2.9 Various picture types in a GOP structure

An inter frame is represented by motion vectors and motion displacement frame. To reconstruct an inter frame, the earlier coded frame is taken and modified according to the motion vectors. This process is called motion compensation. The compensated frame is then added with the motion displacement and the frame is reconstructed. MPEG-2 introduces a concept of group of pictures (GOP)[46] as a coding unit, which consists of various types of frames. Figure 2.9 gives an example of a GOP and the prediction relationships among various frame types.

2.2.3.3 Generic hybrid video coder

The most widely used family of video coding algorithms are called hybrid video coders [64]. They employ motion estimation and compensation to track the frame changes. Meanwhile, they also employ transform coding for image compression. The name "hybrid" comes from this combination. Figure 2.10 shows the structure of a generic hybrid video coder [64]. Here the DCT is assumed to be the transformation method.

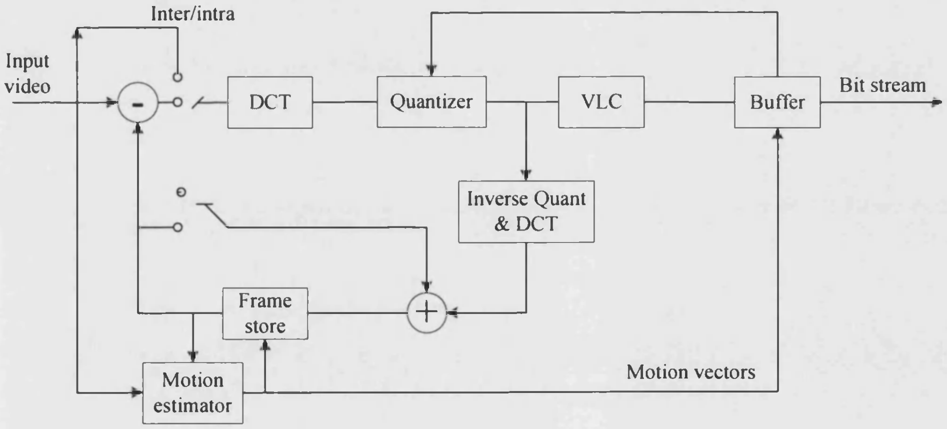


Figure 2.10 A generic DCT hybrid video coder

Compared to motion JPEG, a hybrid video coder is different in that it exploits motion estimation and compensation to make use of the correlation in the temporal direction. It also preserves the three stage image coding structure for both intra frame and displacement coding. Finally, both the quantized coefficients and the motion vectors are entropy coded and sent out with the bit stream.

It should be noted that a motion video coder requires a decoder inside the encoder. In a hybrid video coder, transform coding is performed over displacement frame in which quantization is involved. The information loss during this process must be considered. Since the motion estimation should be based on the reconstructed frame, which is available for encoder and decoder side, the de-quantization must be involved in the motion decoder, as shown in Figure 2.10.

Hybrid video coders have been very successful, and most of the existing coding standards are from this group, including the H.26X family [3, 7, 16, 42, 47] and the MPEG family [5, 15]. The latest H.264 standard [12, 25] exploits an new integer transformation scheme but still uses the hybrid structure of combining motion compensation and transformation.

2.3 Rate distortion measurement and optimization

It has been introduced that in image coding, the quantization part is responsible for discarding part of the information and hence control the ultimate bit rate. To optimize a coder's performance, it is important to manage the quantization part.

2.3.1 Measurement of the rate and distortion.

To study how the rates relate to the distortions, these two factors first need to be measurable, or quantitative. Binary code is normally used for the final output bit streams and therefore the rate is easy to calculate. It is given by,

$$R = Bits / T \tag{2.13}$$

where *Bits* is the total bits number of the file after compression and *T* is the display time the corresponding video is for.

The distortion or the video quality can be measured by a few methods. Psycho-visual model describes the quality based on experiments and statistics of people's ranking. This is a good way but takes a lot of time and experiments so very hard to carry out.

Therefore, more widely used are the mathematical models for measuring the video or image quality, which are both easy to calculate and compare, although highly imperfect. The principle idea of these methods is to compare the reconstructed signals with the original signals and calculate the difference between the two. Following are some frequently used models, including sum of squared distortion (SSD), mean square error, (MSE), sum of absolute differences (SAD), and peak signal to noise ratio (PSNR) [64]. The definitions can be found below,

$$SSD_{\mathrm{A}}(F,G)=\sum_{s\in\mathrm{A}}(F(s)-G(s))^2 \tag{2.14}$$

$$MSE_{\mathrm{A}}(F,G)=\frac{1}{\mathrm{A}}SSD_{\mathrm{A}}(F,G) \tag{2.15}$$

$$SAD_{\mathrm{A}}(F,G)=\sum_{s\in\mathrm{A}}\left|F(s)-G(s)\right| \tag{2.16}$$

$$PSNR_{\mathrm{A}}(F,G)=10\log_{10}\frac{(Peak)^2}{MSE_{\mathrm{A}}(F,G)}decibels \tag{2.17}$$

where Peak designates the peak value of the signal, and in the commonly used portable grey map (PGM) images it is 255. In all equations, *F* and *G* are two array arguments, normally the original and the reconstructed luminance or chrominance values in video coding. Different from the other three metrics, PSNR is a quality measure, and the higher value of PSNR indicates better quality. It is also the most widely used criterion.

The main criticism about PSNR is that the human interpretation of the distortions at different parts of the video can be different. However, so far there are always some issues that even very sophisticated models can not resolve. Meanwhile, it is true in most cases that a video signal with higher PSNR is likely to have better subjective visual quality than one with lower PSNR. Therefore, PSNR is still applied in many occasions. In this thesis, PSNR is the main quality criterion for assessing video quality.

2.3.2 Rate and distortion optimization

It can be seen that the distortion can be reduced to zero if there are enough bits to be used for coding purpose, while the distortion will increase if the number of bits used for coding is reduced. The relationship between Rate and Distortion can be described using a function,

$$D = f(R) \tag{2.18}$$

where R stands for the rate and D designates the resultant distortion, and *f(•)* is called rate distortion function [8, 64].

Every lossy data compression method has only a finite set of admissible quantizers and can produce finite number of output bit rate which makes up a finite number of possible rate distortion pair. Therefore the function describing the rate distortion relationship is discrete, and this is called operational rate distortion theory. The operational R-D theory can be used to guide the coding process. There are also a few algorithms employing this theory to have an optimised coding result, such as Lagrangian multiplier algorithm [64, 65] and dynamic programming [64, 66, 67]. Following the Lagrangian multiplier algorithm [64, 65] is reviewed.

Lagrangian multiplier method is a mathematical tool that solves constrained optimization problems [65]. It is famous for solving constrained minimization problems in a continuous framework, and it also valuable for constrained, discrete optimization problems.

Theorem 1: Let S_B be a finite set and $B \in S_B$ be a member of that set. Let R (B) and D (B) be real valued functions defined over S_B. Then, for any $\lambda \geq 0$ the optimal solution $B^*(\lambda)$ to the unconstrained problem,

$$\min_{B \in S_B}(D(B) + \lambda R(B)) \tag{2.19}$$

is also an optimal solution to the constrained problem:

$$\min_{B \in S_B} D(B), \quad \textbf{subject to: } R(B) \leq R(B^*(\lambda)) \tag{2.20}$$

The theorem just says that to every non-negative λ, there exists a corresponding constrained problem whose solution is identical to that of the unconstrained problem. The main point of the theorem is that it converts a constrained problem into an unconstrained one. The case where $R(B^*(\lambda))$ happens to be an upper bound is first considered and now $R_{\max}$, $B^*(\lambda)$ is the desired solution to the constrained problem:

$$\min_{B \in S_B} D(B). \qquad \text{subject to: } R(B) \leq R_{\max} \tag{2.21}$$

Thus the constrained problem is converted to an unconstrained problem. What remains to be solved is to find a λ such that $R(B^*(\lambda)) = R_{\max}$. Considering this problem in video coding, $D(B)$ is the distortion for encoding a given source using a certain quantizer B and $R(B)$ the corresponding rate.

Theorem 2: If $R(B^*(\lambda_1)) > R(B^*(\lambda_2))$, then

$$\lambda \geq -\frac{D(B^*(\lambda_1)) - D(B^*(\lambda_2))}{R(B^*(\lambda_1)) - D(B^*(\lambda_2))} \geq \lambda_1 \tag{2.22}$$

This theorem states that given two optimal solutions, the ratio of the change in optimum distortion to the change in the required rate is bounded between the two multipliers. In particular, if the set of solutions produced by Lagrangian multipliers results in a distortion that is a differentiable function of the rate at some point, then it follows from the above theorem that λ at that point is the derivative of the distortion with the respect to the rate given by:

$$-\frac{dD}{dR} = \lambda \tag{2.23}$$

This property will be further discussed in Chapter 3. Theorem 2 assures that a particular λ value lies between a higher and a lower bound of λ values. Following this, iterative bisectional search is often used to find the target λ which is of big practical importance.

2.4 The unified video coding framework

2.4.1 The application of framework in video coding research

A video framework refers to the code structure that can be used to build up a video coder. Normally it needs to have proper interfaces so that various coding blocks can be easily adopted. Any video coding technique can be applied to a video framework to verify its effectiveness.

All the video coding standards described in the last chapter are hybrid video coders which are built on the approach involving block-based motion compensation and DCT based frame coder. The motion compensation tries to reduce the temporal redundancy by tracking the motions between adjacent frames. Unlike motion JPEG, which codes each frame independently as described in section 2.3.2.1, motion compensation exploits successive prediction from frame to frame. Figure 2.11 depicts such a relationship. I_0 and I_1 are two adjacent frames. And I_0^R is the reconstruction of I_0. I_1^P is the prediction of I_1, which is obtained from I_0^R. Therefore, any change that could affect an earlier coded frame's quality might cause changes among successive frames as well.

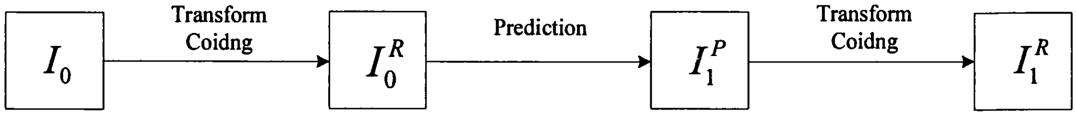


Figure 2.11 Prediction relationship between adjacent frames

Such changes would propagate along the video sequence. So if we want to modify certain part of the video system and find out its impact to the ultimate result, a video framework with motion prediction model has to be employed. For the purpose of comparison, several popular and major video standards, such as MPEG-2 [43, 46], H.263 [42, 44, 47], and H.264 [26, 27, 50, 68], are most commonly used as framework. This approach is taken in this thesis as well.

2.4.2 The unified video framework

A unified video framework was developed based on H.264 reference software. This framework provides diverse functionalities and meets a wide range of requirements in research work.

2.4.2.1 Converting the H.264 reference software into a video framework

The H.264 reference software is available from [69]. At the time when this thesis is written, the latest version of the software is JM11.0, which was updated in Aug 2006. The video framework was developed from JM9.2, which was the latest available version of the code at that time.

JM9.2 provides all the major distinctive features of H.264, and some advanced functionalities, such as rate control, fast motion estimation, etc. JM9.2 provides a good coding efficiency and is a good choice to be used for developing the video framework. This framework is applied in some of the experiments in later chapters.

2.4.2.2 The versatility of the framework

This framework can be used to build a video system retaining some of the features from H.264, meanwhile also adopting some other coding modules. It provides diverse functions that are often fancied in related video coding experiments.

A. Dynamic intra image coder

In this case, interface for intra image coder is created and other still image schemes with the same input and output parameters can be fitted in. This can be used to evaluate different intra frame encoders. In the case when a new intra image coder is employed, the H.264 intra image coder is skipped.

With proper parameter settings, a video coding in a fashion similar to motion JPEG can be carried out. With this framework, it can be done in two ways. First, this can be done by choosing to set a sequence to be only intra frames, and coded by H.264. Second, replace the H.264 intra frame coder, and apply the same settings as in the first case, then the frames of that video sequence can be coded independently by the new image coder.

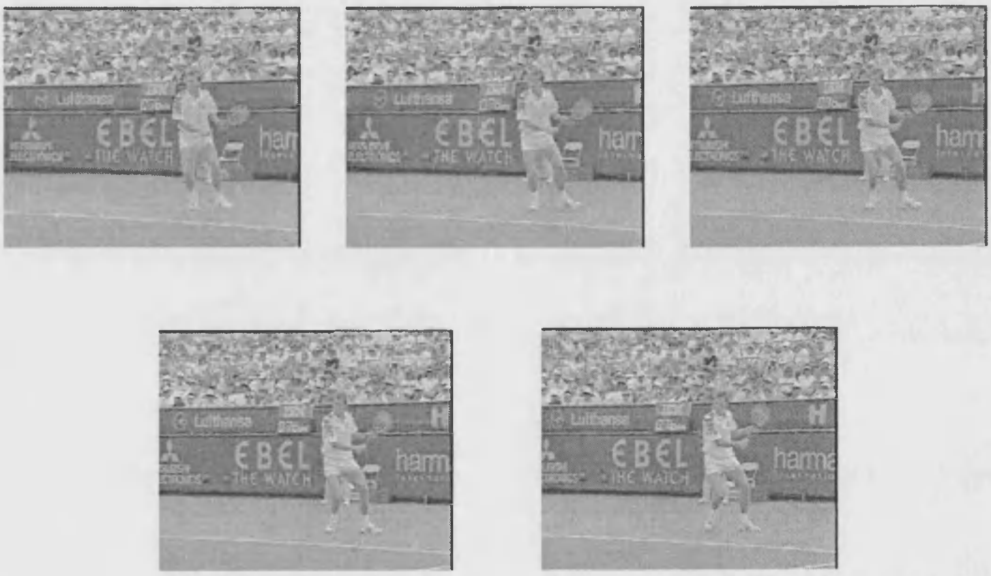


Figure 2.12 The first 5 frames of Stefan coded by H.264 intra mode.

In Figure 2.12 and Figure 2.13 two examples of such coding results are presented. In Figure 2.12, the first 5 frames of Stefan are coded by the Intra mode of H.264. The same frames are coded by JPEG2000 and the reconstructions are shown in Figure 2.13. This can be regarded as a motion JPEG2000 algorithm in comparison with MJPEG [62].

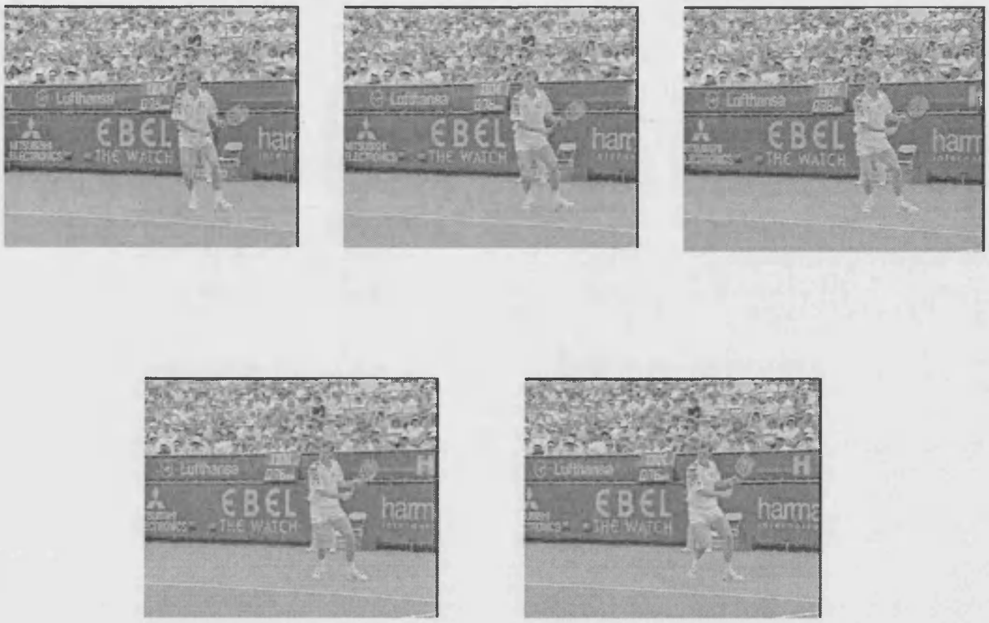
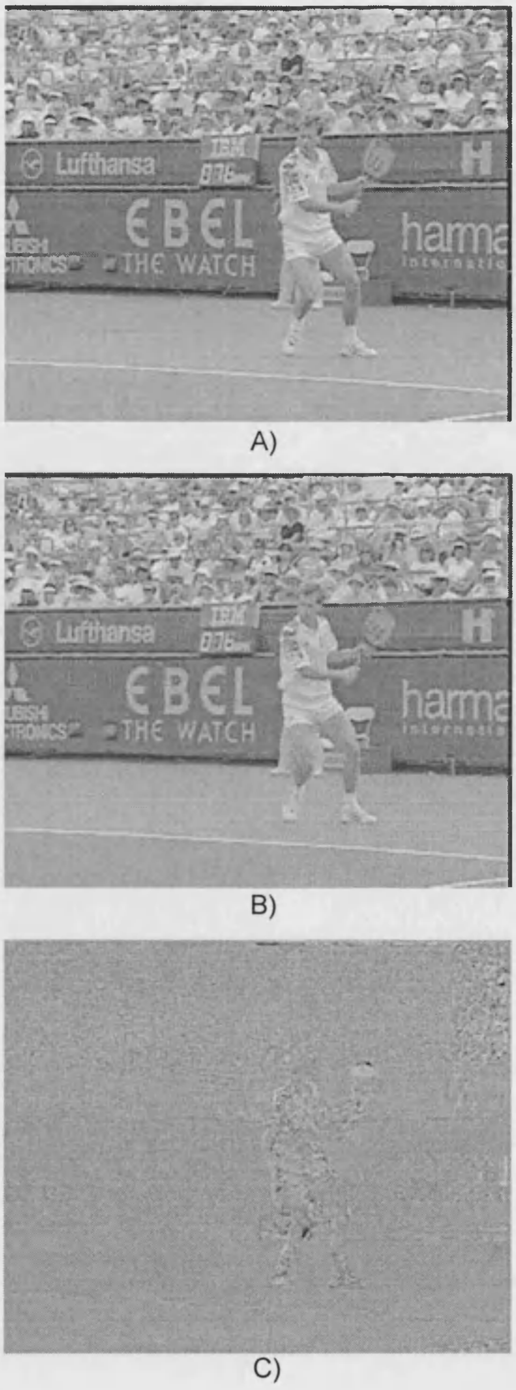


Figure 2.13 The first 5 frames of Stefan coded by JPEG2000.

B. Motion compensation frame data extractor

As one of the most important parts in a hybrid video coder, motion compensation produces some different data during the encoding process. To help carrying out research on inter frame coding, the framework is designed to extract these data.

H.264 follows a traditional macroblock based frame coding fashion, and therefore, to make a copy of the data for an inter frame, the values must be saved each time the motion prediction is carried out over a macroblock. The framework also provides an optional function of writing that residual frame into file so that the data can be exported. This function would be useful for research on matching pursuit encoding, which is presented in later chapters. Figure 2.14 displays examples of some different types of frame data that can be extracted from the framework.



A)

B)

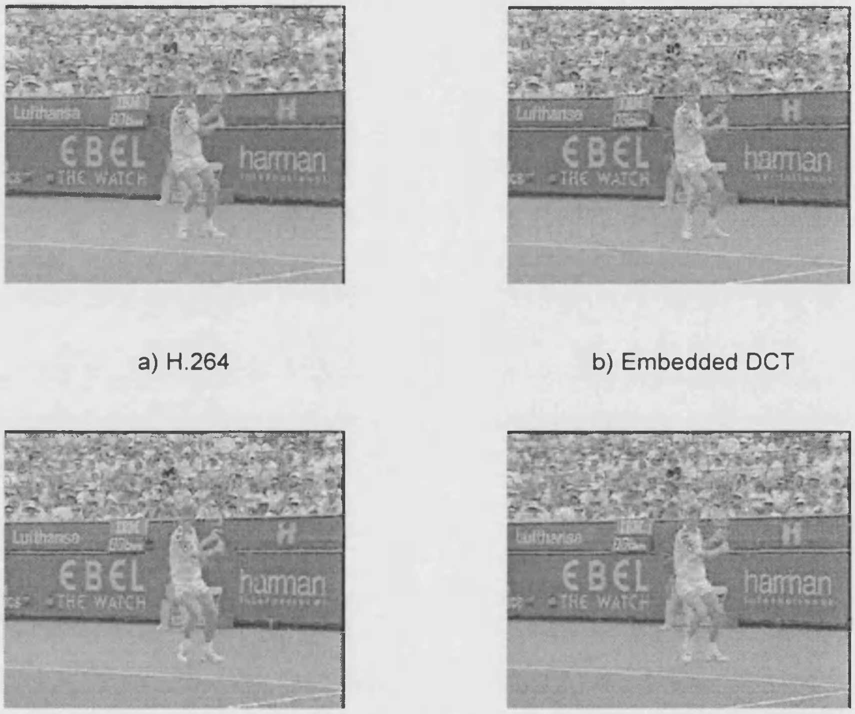
C)

Figure 2.14 Different data 5th frame of Stefan:

A) Original Frame; B) Motion Prediction; C) Residue Frame

C. Flexible displacement image coder

The inter frame encoder of H.264 can be divided into motion estimation part, which includes the motion vector coding, and the displacement residual coder. Now with all the motion compensation related data available, we can make the displacement frame encoder replaceable. A specific interface is also made for the displacement frame coder that would connect any coding scheme with proper parameters. The saved residual frame can be fed to the adopted new encoder. This functionality is depicted in Figure 2.15. Four different displacement coders are used to code the ninth frame of Stefan and the reconstructed frames are presented.



a) H.264

b) Embedded DCT

c) JPEG2000

d) Bath- University matching pursuits

Figure 2.15 The 9th frame of Stefan coded by different algorithms

D. Bit usage monitor

Specific variables are allocated in the framework to record the bits usage for encoding various types of data. In a hybrid video coder, a video signal is separated into different types of information, including motion vectors, reference frame number, motion prediction modes, quantized coefficients, etc. All these are to be encoded for transmission. It is important to record the bit numbers used for various types of information and use them in rate calculation and controlling.

For the purpose of comparison, it is often required to carry out coding at the same bit rate by different encoders. Thus, when adopting a new intra image coder into the system, the bits used for sequence header should not be given as part of the bits for encoding that intra image. For an adopted displacement residual coder, the bits used to encode the motion vectors, and motion prediction modes can not be used for displacement coding. With the bit usage monitor mechanism, the proper bit budget for each part can be calculated and applied.

E. Retained motion model

The advanced and complex motion model helps H.264 to gain the superior coding efficiency. This thesis does not look into motion estimation and compensation, therefore this motion model is retained in the framework. With the changes above, all the advantages of the model are reserved, including the high resolution, multi reference frame, various block shapes, etc.

The structure of the framework is depicted in Figure 2.16. The dashed rectangles highlight the unique features described above.

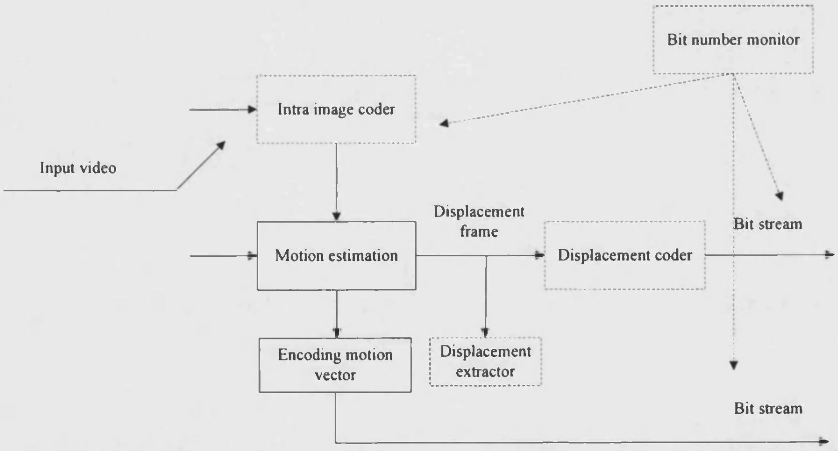


Figure 2.16 The proposed video framework

The reference software from [69] is written in C language. Considering that many other modules that might be fitted in were developed in C++, this code is converted into a system by C++ language. This also introduces the idea of object oriented programming into the framework, and assures encapsulation in the code structure, which is good for design of later diverse interfaces, shown in Figure 2.17.

The different modules are integrated into a single object of "Generic-Coder", which acquires all necessary information through some specific interfaces. All the necessary feedback of data, such as reconstructed frames, bit usage, etc, is also performed through some interfaces. The data extraction is performed inside the Generic-Coder and is not shown in the figure. Various optional intra and inter image coders are connected to the Generic-Coder without communicating with the H.264 code, as shown in Figure 2.17. And the calculation is done inside the object without interfering with the memory used by the original H.264, which prevents a lot of possible mistakes in data usage.

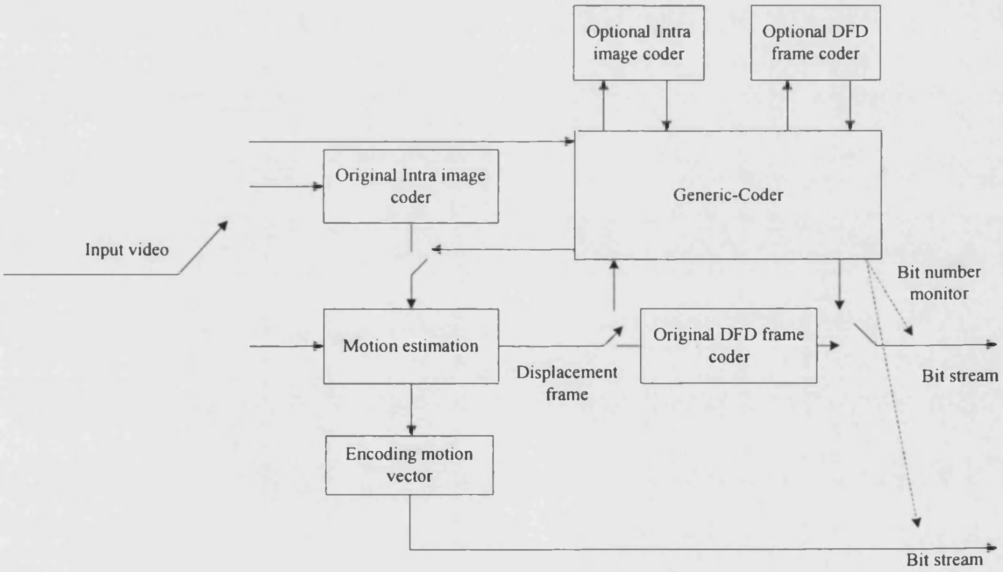


Figure 2.17 The framework with encapsulated generic coder

2.5 Summary

In this chapter, we have reviewed the main techniques involved in image and video compression, each of them focusing on removing the inherent spatial and temporal redundancy of a video sequence. Some international standards for image and video compression have been introduced. All these standards employ certain techniques and have their own specific features. The problem of rate distortion measurement and optimization is then addressed. In the final section, the versatile video framework that is used in the later part of this thesis was outlined. The framework is made with object oriented system structure based on H.264 standard reference software, retaining its advanced motion model and providing some useful features for research work

Chapter 3 RATE DISTORTION OPTIMIZATION FOR H.264 WITH QUANTIZATION PARAMETER PICKING

In the last few decades, a series of video coding standards have been released since the 1990s [37]. Along with these developments, more and more features were introduced to meet all sorts of applications. Meanwhile, various details of the basic approach were refined and resulted in more complex and also more efficient compression standards. H.264 introduces a lot of advanced features which enable it to give an excellent coding efficiency. It has triggered a lot of interest and will no doubt soon fit into a lot of practical applications. The block based style hybrid video system is still going to be used in a wide range for a long period of time. Therefore, it is worth looking into the system and find how its efficiency could be improved with rate control.

3.1 H.264 in a nutshell-new features and new concerns

ITU-H.264 [27, 35, 50, 70] is the latest video standard and holds a lot of features that enable it to be applicable to a wide range of application domains, including adaptation to delay constraints, error resilience, network friendliness, and so on. In addition, H.264 is also presently the most advanced standard in terms of compression efficiency [71]. H.264 retains the basic coding style of H.26X family, and such a good performance gain occurs in the details of each functional element. Following are some of the distinctive features that make H.264 more efficient comparing to other standards:

- Advanced motion vector accuracy and more sophisticated motion models. In the H.264 (MPEG-4 AVC) standard, more precise motion compensation prediction: up to 1/4 and 1/8 pixel precision, while in H.263, only half pixel precision is allowed. It also employs MC blocks of variable sizes and shapes, so that motion in a frame can be more precisely tracked. H.264 even allows the storage of multiple frames of the past for temporal prediction, and the encoder has the chance to choose a better frame to predict from [27, 35, 50, 70] .

- H.264 uses an integer based transformation that approximates the DCT used in previous standards. The coefficients produced in such transform are all integers, and this helps to prevent decoder side mismatch problem, and avoids the problem of overflow [72].

- In order to improve the compression efficiency of coding Intra frame, an intra prediction scheme is introduced in H.264, which helps to further exploit the correlation among adjacent transform blocks. As an option, variable-sized transformation is also allowed to exploit [73].
- A very effective context based adaptive binary arithmetic coding (CABAC) is exploited [74]. Due to the novel context model, it provides excellent compression [74]. A context adaptive variable length coding (CAVLC) function can also be chosen to perform entropy coding [27, 35, 50, 70].
- Advanced adaptive deblocking filter is applied. The deblocking filter is adaptive according to the quantization step for each macroblock. It not only removes the unpleasant blocky effect, but also improves the PSNR [75].

A coding system with all these features defines the best coding efficiency in literature and is worth of being utilized as a benchmark for research purpose. However, such new features also raise the necessity of extra concern to be taken when designing corresponding rate control scheme.

3.2 Quantization scheme for H.264

In H.264, like the other video coding standards, the strength of quantization is controlled by a specific parameter. In this chapter, the process of choosing the quantization parameter and how it could affect the final video quality is looked at in depth.

3.2.1 The combination of transformation and quantization

In H.264, a special parameter called the quantization parameter (QP), is reserved for controlling the strength of quantization. As a parameter, QP is reserved for the same purpose in the past video standards such as MPEG-I,-II [2, 34, 46, 52, 76] and H.26X [34, 44]. In H.264, the concept is further developed.

Compared to earlier video standards, differences of H.264 mainly exist in the way how quantization is implemented. In H.264, the quantization and transformation are combined together and implemented within the so called scaling and transformation operations [72].

In H.264, an integer transformation scheme is employed [73]. The transform matrix involved is set to simulate the DCT but rounded to integer values. The transform matrix is given in equation (6.1). The coefficients of DCT of same size in equation (6.2) and the integer transform below are compared.

$$H_{\text{int}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{3.1}$$

$$H_{\cos} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1.3 & 0.54 & -0.54 & -1.3 \\ 1 & -1 & -1 & 1 \\ 0.54 & -1.3 & 1.3 & -0.54 \end{bmatrix} \tag{3.2}$$

Such a transformation scheme has a few advantages. The 4x4 transformation size is small and thus reduces the blocking effect. The transformation can be calculated with only integer arithmetic, thus avoiding the inverses-transformation mismatch problem. Also, the transformation coefficients are significantly simplified. The matrix only contains numbers of ±1 and ±2, and the calculation involves only addition and bit shifting, which enables much quicker computation, and saves processing power at the encoder.

However, there is an obvious deviation between the Integer transform and the Cosine transform coefficients. The integer coefficients are rounded in a very coarse way. In order to compensate this effect, the following scaling factors are set differently to each frequency coefficient. The scaling factors table, represented by V is shown below. The actual factor chosen from this table for a coefficient depends on both the predefined QP value and its frequency within the transformed matrix.

$$V = \begin{bmatrix} 10 & 16 & 13 \\ 11 & 18 & 14 \\ 13 & 20 & 16 \\ 14 & 23 & 18 \\ 16 & 25 & 20 \\ 18 & 29 & 23 \end{bmatrix} \qquad (3.3)$$

3.2.2 The quantization steps

For the purpose of better accuracy and easier calculation, H.264 introduces combined transformation and scaling scheme, as addressed above. Yet from a user's point of view, this scheme achieves similar result as earlier video standards do, and can be controlled with the parameter, QP. The actual implementation process can be skipped when considering the effect of quantization. In the discussion of this chapter that follows the problem is only considered externally.

Compared to earlier standards, the QP is allowed to take more values. In earlier standards like H.263 and MPEG-II, QP should be within the range [0,31], while in H.264 the range is extended to be [0,51]. Obviously this allows wider control options. Therefore, the H.264 coefficients can be coded much more coarsely than those of H.263 and other standards for higher compression and more finely to produce better quality, if the bit budget allows.

Another interesting character of this scaling scheme is that the scaling equations are specified such that the equivalent scaling parameter doubles for every increment of 6 in QP. Thus, there is an increase in scaling magnitude of approximately 12% from one QP to the next.

In accordance to the needs of the application of QP, H.264 also provides another parameter to accompany it. Since the entropy of a video signal varies with the content, effective usage of bits is very important in order to achieve a good coding result. Changing bits number has to be accomplished by changing the quantization strength, or, changing QP. In H.264, delta-QP is used to change the QP value. It conveys the difference between the original and the new QP values. Delta-QP can be used at any level. It can be used to change QP at a new frame or, a new slice or even a new macroblock. The new QP values are obtained by computing [69]

$$QPY = (QPY,PREV + mb\_qp\_delta + 52) \% 52 \qquad (3.4)$$

In this way, QP can be changed at both slice level and macroblock level.

3.3 Difficulty for rate control in H.264

In order to achieve a higher coding performance, a rate distortion optimization (RDO) algorithm [27, 71] is proposed in H.264. Encoding modes and motion estimation models have been made much more complicated in H.264 than in earlier standards. Decisions of choosing from those candidate modes are made by RDO, in which an empirical Lagrangian multiplier value is allocated according to the QP applied to that MB. The encoding modes or block shape that minimizes the encoding cost function are chosen.

Such a scheme helps to choose the best coding modes during encoding, but also makes a challenge to rate estimation and control. The model[28] exploited in rate control needs to calculate the quantization parameter (QP) from the distortion to allocate the target rate, while the QP must be first decided before the modes and distortion can be decided. This is referred as chick and egg dilemma[29]. In [77], an algorithm avoids this problem by switching RDO off and focusing on the buffer status. This would obviously reduce the system performance. One well known existing scheme was proposed by Ma[78], and has been adopted in JM92. It tries to predict the QP value and apply a two pass scheme to solve this problem. This obviously lacks accuracy. An algorithm able to avoid such problem would potentially improve the ultimate coding efficiency.

3.4 Rate control with operational quantization parameter picking

Allocating different number of bits to various parts of a video signal can be realized by applying different QP values over different parts. Since the entropy of a video signal varies due to the content, effective usage of bits is very important in order to achieve a good coding result. The basic purpose behind this is to allocate the bits to code the part of information that would result in most distortion deduction. This problem can be solved with some optimization methods, as described in Chapter 2. The Lagrangian multiplier exploited in H.264 optimizes the modes and other encoding options, but does not consider the ultimate bit rate after the residual is coded. In the chapter, the problem of picking QP and then optimize the coding settings is to be studied.

3.4.1 Slope matching scheme

3.4.1.1 Physical meaning of Lagrangian multiplier

To achieve good coding quality, some optimization methods such as Lagrangian algorithm are often exploited. The Lageangian multiplier method can be briefly described below.

A cost function is defined to measure the R-D performance of a selected parameter setting:

$$J = D + \lambda R\text{,} \quad (R = \sum R(n)\ D = \sum D(n)) \tag{3.5}$$

where D(n) is the resultant distortion when coding the nth unit, and R(n) is the bits consumed for coding the nth unit, and λ is called Lagrangian multiplier. The process of optimization is to find a proper R so that, together with R the resultant D minimizes the cost function J, that is

Optimal coding is achieved when

$$MIN(J = D + \lambda R) \tag{3.6}$$

is obtained. The chosen optimal R changes with the Lagrangian multiplier λ and so λ can be changed to make the coding meet the target bit rate.

For each coding unit I, the point on the R-D characteristic that minimizes J is that point at which the line of the absolute slope λ is the tangent to the convex hull of the R-D characteristic [64], i.e.,

$$\frac{\partial J}{\partial R} = \frac{\partial D}{\partial R} + \lambda \frac{\partial R}{\partial R} = 0\text{,} \qquad \lambda = -\frac{\partial D}{\partial R} \tag{3.7}$$

For this reason, λ is often referred to as R-D slope, which describes the changing trend of the relationship between R and D. Since λ is a constant in all the coding units of the sequence, the Lagrange multiplier optimization is also called constant slope optimization.

3.4.1.2 Unifying the R-D slope

The principle behind the Lagrange multiplier is to achieve a constant Rate-Distortion slope in a certain range, such as macroblocks within a frame, or macroblocks and frames within a group of pictures, and such that the optimal coding results can be obtained.

Based on this idea, an optimization process can be carried out in the following way: the slope for a unit can be calculated by applying various QP values and by adjusting the QP of each unit the slopes can then be made close to a target value. In general the slopes can be unified. In this way, a superior general quality should be obtained.

3.4.1.3 Slope unifying control scheme: difficult in practice

Following the idea of improving the whole coding performance by unifying the R-D slope of each coding unit, the Slope Unifying scheme which adjusts the R-D slope to match a constant R-D slope value is proposed. The scheme can be described by the block diagram in Figure 3.1.

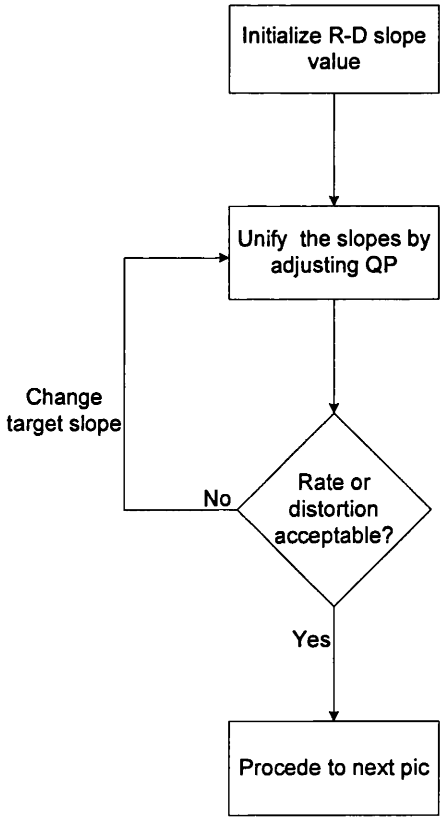

Initialize R-D slope value
Unify the slopes by adjusting QP
Change target slope
No
Rate or distortion acceptable?
Yes
Procede to next pic

Figure 3.1 Slope unifying control scheme

A. Deciding the target slope

Since this scheme aims at modifying coding settings to match the target slope, the slope value must be established first. As the only standard in the scheme, the slope value determines the final coding settings to be chosen along with the resultant rate and distortion. Various targets can be assigned to one video sequence, which would lead to different coding bit rates. There are a few factors that need to be taken into account when deciding the target slope:

[i]. Target coding bit rate.

[ii]. The GOP format and the image type.

[iii]. Sequence content and the frame rate.

The relationship between the slope value and the conditions mentioned above is investigated in experiments.

B. The characters of slopes

The QP values, from the lowest to the highest, represent the finest to the coarsest quantization. Coarser quantization leads to larger distortion, and smaller compression size, or

$$SAD(i) \leq SAD(i-1), \quad i \in (Q_{\min}, Q_{\max}] \tag{3.8}$$

The R-D slope refers to the relationship between the distortion and the bits consumed, which can be stated mathematically as,

$$D'(R) = \frac{dD}{dR} = \frac{dD/di}{dR/di} \approx \frac{D(i)-D(i-1)}{R(i)-R(i-1)} = \frac{\Delta D}{\Delta R} \tag{3.9}$$

and this value decreases monotonically with R. This property is empirical and has been validated in numerous experiments [18]. This property is exploited in this new control scheme.

C. Searching the target slope- the difficulty

The second block in Figure 3.2 is the key part in the scheme. At this stage, the R-D slope that is closest to the target value is searched, and then the QP value which lays the R-D result along the slope is located. This QP will then be applied. To investigate on the improvement, QP is changed at macroblock level. The process of tuning is illustrated in the diagram below.

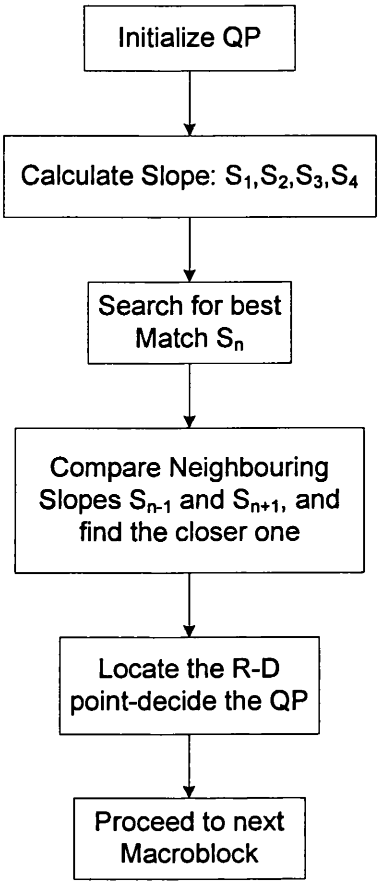

Initialize QP
Calculate Slope: S_1, S_2, S_3, S_4
Search for best Match S_n
Compare Neighbouring Slopes S_{n-1} and S_{n+1}, and find the closer one
Locate the R-D point-decide the QP
Proceed to next Macroblock

Figure 3.2 The procedure for searching the best slope

One R-D slope is defined by two points on the R-D plane. Therefore a few candidates R-D points need to be found. When the QP values are being modified, it is widely accepted that the changes should not be larger than 2 between the two successive units [18]. In terms of mathematics, let the QP of the kth coded macroblock be Q(k), then the QP range for the coded frame at the (k+1)th stage can be limited as:

$$Q(k+1) \in [Q(k)-2, Q(k)+2] \cap [Q_{\min}, Q_{\max}] \tag{3.10}$$

where $Q_{\min}$ and $Q_{\max}$ are the minimum and maximum QP supported in the video coding method. In H.264, they are 0 and 51 respectively.

So, there are at most five candidate QPs to be tried out for slope searching, (i.e., QPPREV -2, QPPREV-1, QPPREV, QPPREV+1, QPPREV+2). A macroblock is first coded by the 5 (or fewer) different QP settings, and 5 R-D pairs are found (see Figure 3.3). With these five points, 4 slopes can be obtained. By comparing the values, the slope S_n which is closest to the target value is found. The two R-D points defining this slope are taken as candidate QPs. One R-D point is to be decided with two slopes. So, S_n's two neighbouring slopes, S_{n-1} and S_{n+1} are compared. Of the two values, the one closer to the target is picked. Together with S_n, it locates the QP that is needed.

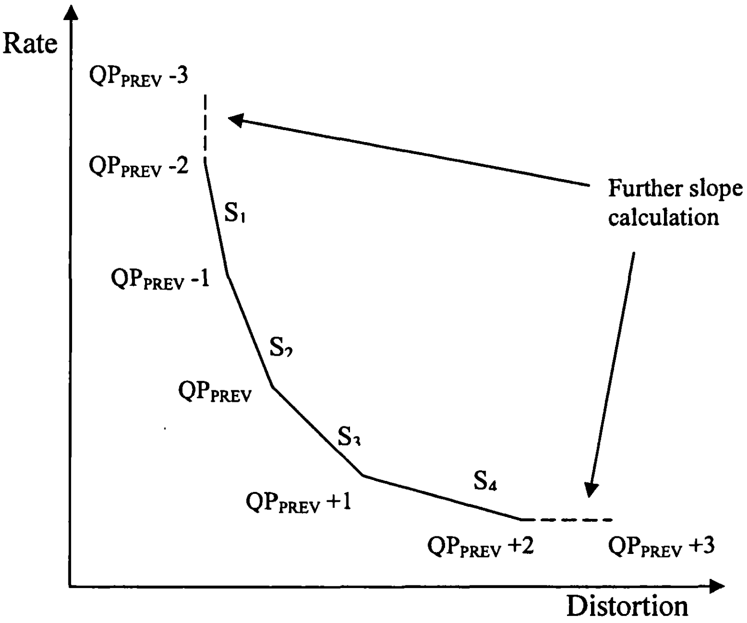


Figure 3.3 Slope calculation and searching

It should be noticed that if S1 or S4 is picked as the best slope, it then has only one neighbouring slope. A further QP value needs to be tried in calculating another slope for QP picking.

In H.264, the QP value not only controls the quantization strength but also affects the motion estimation. An optimization scheme is introduced to decide the mode. The motion estimator considers the QP when comparing the various motion prediction modes. This allows the possibility of changing the number of bits for motion vectors when using various QPs, and the total bit rate for this macroblock can then be changed accordingly, not necessarily monotonically.

This empirical property of the slopes has been analysed: they monotonically decrease with the rate. This property is true at a larger scale, e.g, a frame, or a GOP. But it may not be true for a single macroblock. We show a few slope changes in Figure 3.4.

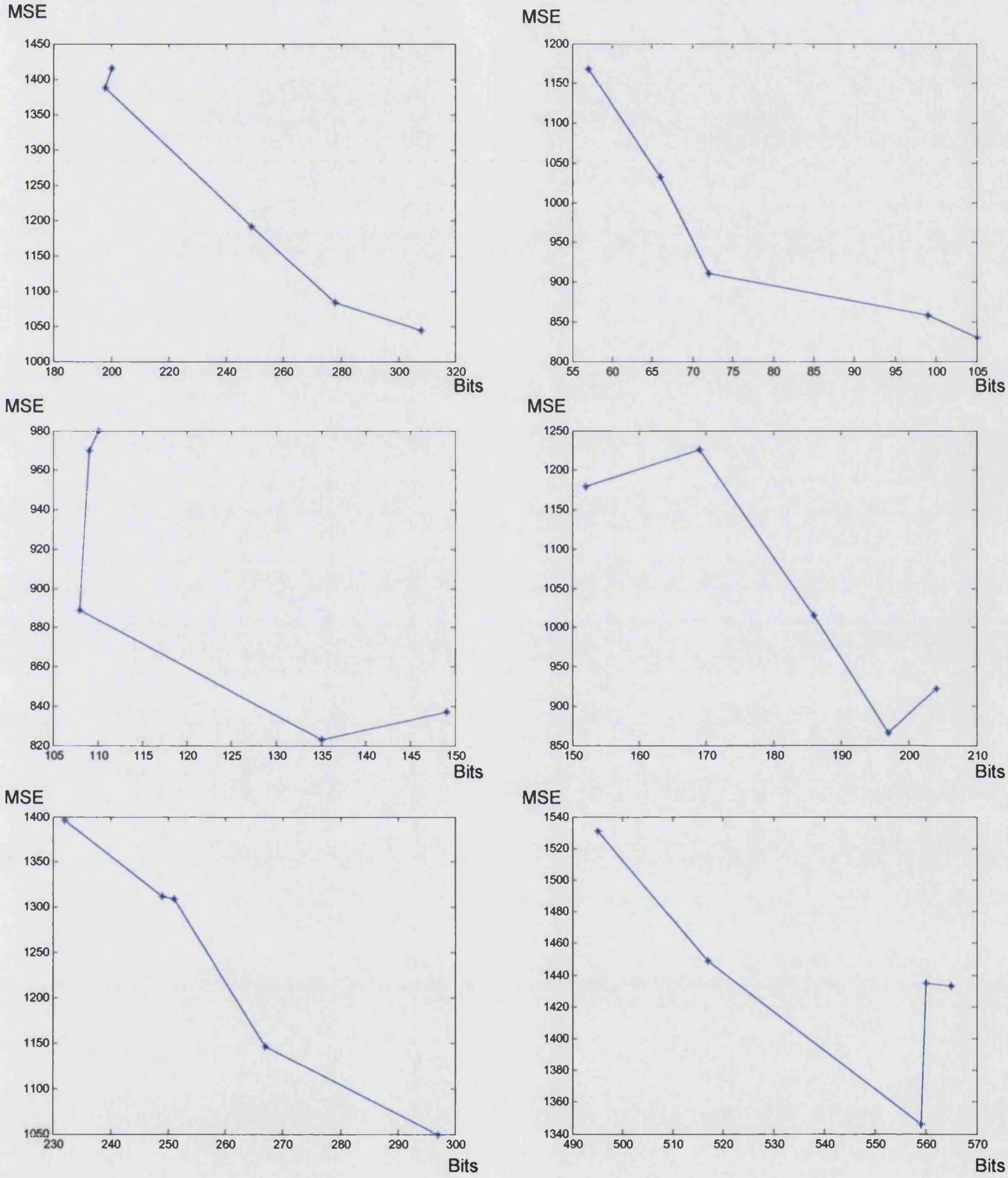


Figure 3.4 The MSE-Rate relationship of several MBs from coding Hall

It is obvious that the slope jumps dynamically with hardly any regularity and such irregular changes in property make the slope searching very difficult. The reasons are:

a). Since the rate and the distortions are not monotonically changing according to the parameter QP, we need to rearrange the sequence to measure the slope. We propose to sort the QPs according to the rate values in a descending order. R-D slope is obtained by calculating dD(n)/dR(n).

b). with different slope value, different resultant bit rate can be achieved. According to the theoretical R-D curve, smaller slopes correspond to higher bit rate and lower distortion.

c). In P frames, motion estimation and compensation removes a lot of temporal redundancy so that a large area of each frame may contain very small values of coefficients and the corresponding MBs can just be marked as skip, and no bits are allocated. In that case, we choose to use the QP that gives the smallest distortion/ no change.

d). In adjacent frames, the chosen QP can be same, so keeping a copy of QPs chosen by previous frames could save time in searching QP.

3.4.2 Quantization parameter picking with Lagrange multiplier

The approach of slope matching has difficulty in practice, as indicated above, and another way need to be found. In this case, the Lagrangian multiplier is reconsidered.

3.4.2.1 Function of coding cost

The problem of achieving the best overall spatial quality can be expressed mathematically as,

$$\min \sum_{i=1}^{n} D_i(Q_1,...,Q_i) \quad \textbf{subject to} \sum_{i=1}^{n} R_i(Q_1,...,Q_i) < B \qquad \textbf{(3.11)}$$

A coding cost function, which has been mentioned in the last section, is introduced in the Lagrangian multiplier algorithm. It is defined to measure the R-D performance of a selected parameter setting with a Lagrangian parameter λ:

$$J = D + \lambda R, \quad (R = \sum R(n) \; D = \sum D(n)) \qquad \textbf{(3.12)}$$

where D(n) is the resultant distortion when coding the nth unit, and R(n) is the bits consumed for coding the nth unit, and λ is called Lagrangian multiplier.

3.4.2.2 Application of Lagrangian multiplier algorithm

The process of optimization is to find a proper R so that, together with R, the resultant D minimizes the cost function J. Optimal coding is achieved when

$$MIN(J = D + \lambda R) \qquad (3.13)$$

is satisfied. By doing this, the problem is converted into one without limitation. The Lagrangian parameter λ now needs to be changed to meet the target rate.

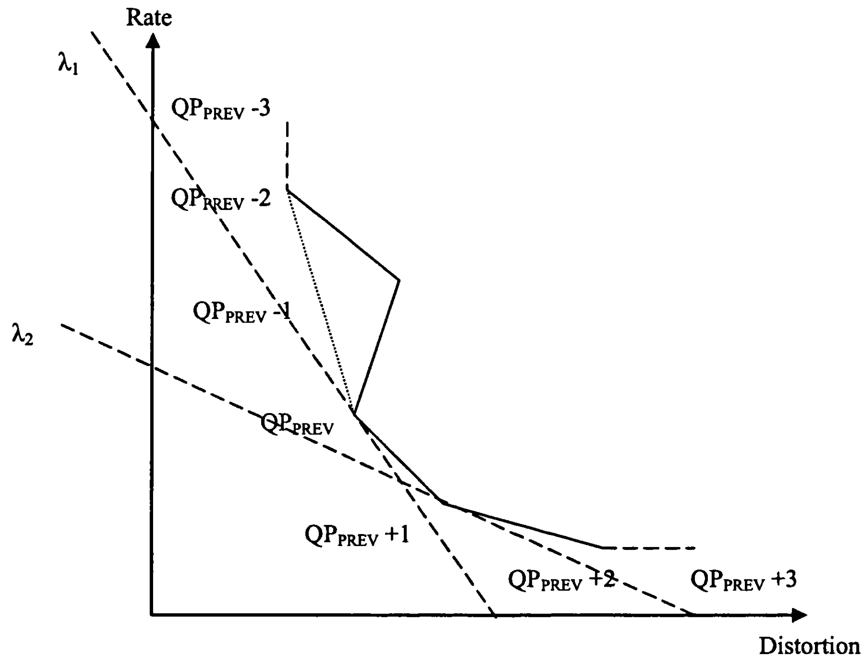


Figure 3.5 Selecting coding setting with Lagrangian multiplier

From the Figure 3.5above, it is obvious that the irregular R-D slope trend would not be a major obstacle for QP picking. During the searching process, no matter what value λ takes, it is made sure that the convex of the rate-distortion curve is traced, so only the R-D points on the convex would be chosen. Additionally, since the problem is without limitation now, the coding cost is the only standard for the picking process. The point giving the smallest cost value is chosen. This makes the picking straightforward and simple.

Now the rate control scheme is modified into the structure shown in Figure 3.6. This approach is similar to the slope unifying approach, but due to the advantage of

Lagrangian multiplier method, is much easier to calculate and implement. So it is good to be applied to the optimization process in practice.

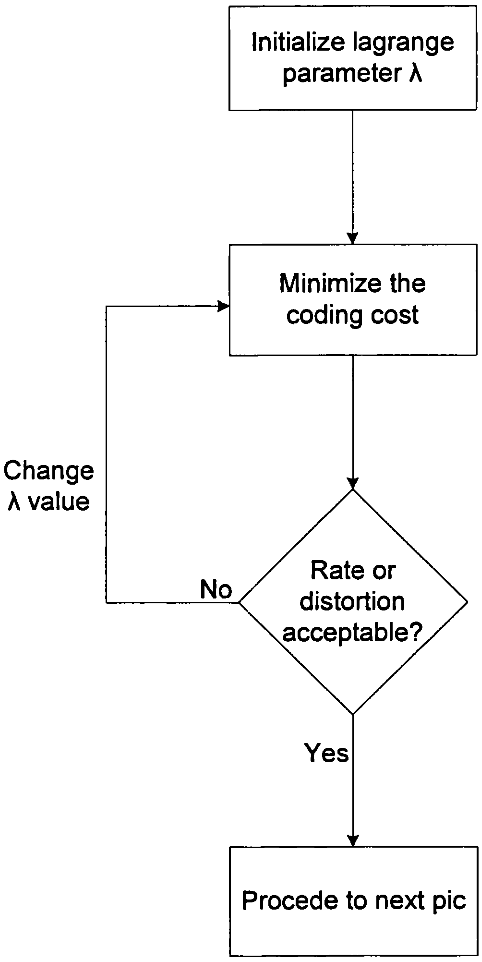


Figure 3.6 The proposed rate control procedure

3.5 Experimental results

The proposed rate control scheme has been implemented and adopted in the video framework.

3.5.1 Implementation details

The block diagram in Figure 3.6 shows that the proposed scheme is based on the optimization scheme of Lagrangian multiplier. The QP value is changed and, hence the bits allocation gets modified.

The H.264 allows the QP value to be changed at slice level or macroblock level. In order to investigate the extreme potential of improvement, this QP picking over macroblocks is applied which is the lowest possible level. Similar to the slope unifying scheme described earlier, for each single macroblock, we apply 5 candidate QP values are applied. The coding cost function for each QP is then calculated. The one that provides the smallest cost is chosen.

It has been mentioned in previous works that dramatic QP changes could lead to high quality jumps, and therefore, might cause flickering effect. The number of candidate QP values is limited to five, making the changes of QP between two adjacent units equal or smaller than two. Also this limit makes sure the complexity of searching is not too high.

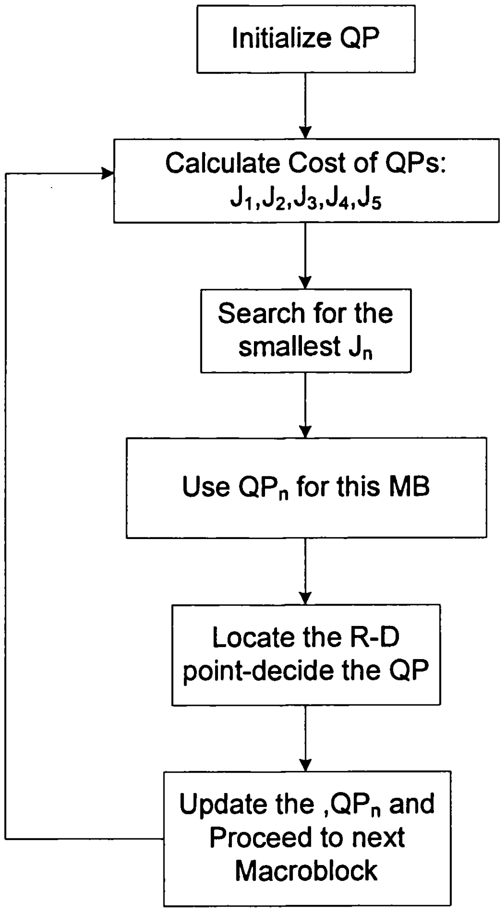

Initialize QP
Calculate Cost of QPs: J_1,J_2,J_3,J_4,J_5
Search for the smallest J_n
Use QP_n for this MB
Locate the R-D point-decide the QP
Update the ,QP_n and Proceed to next Macroblock

Figure 3.7 Lagrange searching Procedure scheme

In the implementation, the delta-QP for signalling the QP changes is used. For each single macroblock, it is coded by 5 different QP values, namely QPPrev-2, QPPrev-1, QPPrev, QPPrev-1, QPPrev+1, QPPrev+2. The number of bits used and the resultant distortion is recorded. Then, five coding cost function values are calculated. The QP value that provides the lowest coding cost is then chosen for use. The sixth time of coding is then carried out with this value for the sixth time. This QP is also used as the QPPrev for the next macroblock (see Figure 3.7).

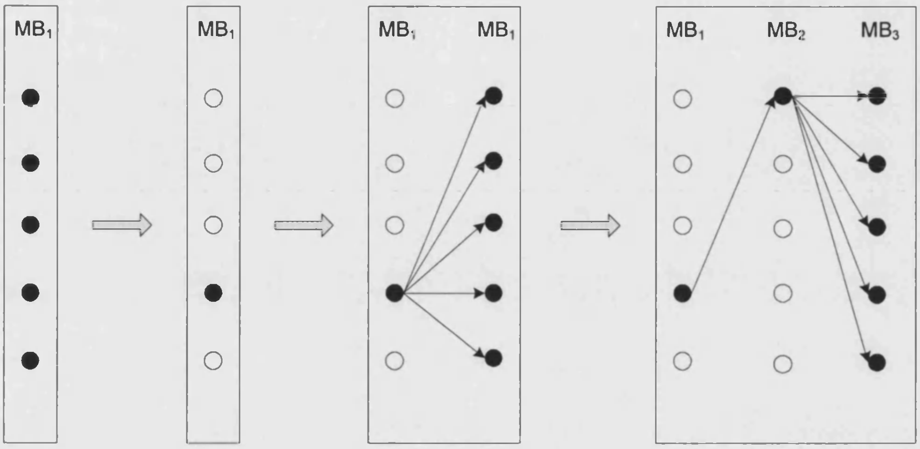


Figure 3.8 The QP picking processing among macroblocks

The picking process can be depicted by Figure 3.8. As can be seen, the process can be regarded as a path finding problem. Each MB can be regarded as a coding stage where the direction of the path needs to be changed. For each stage, the direction of the path is to be chosen from the up to 5 choices according to the coding cost function.

3.5.2 Rate- λ relationship

This scheme guides the coding with a Lagrangian parameter λ and the ultimate coding rate and quality both depend on this value. From Figure 3.9, to Figure 3.14show the relationship between the parameter λ and the bit rate and between λ and PSNR in our experiment.

Generally, the rate and quality decreases as the λ value goes up. The relationship curve changes with the content. If the sequence needs to be coded at a specific bit rate or quality, more iteration must be taken to achieve this.

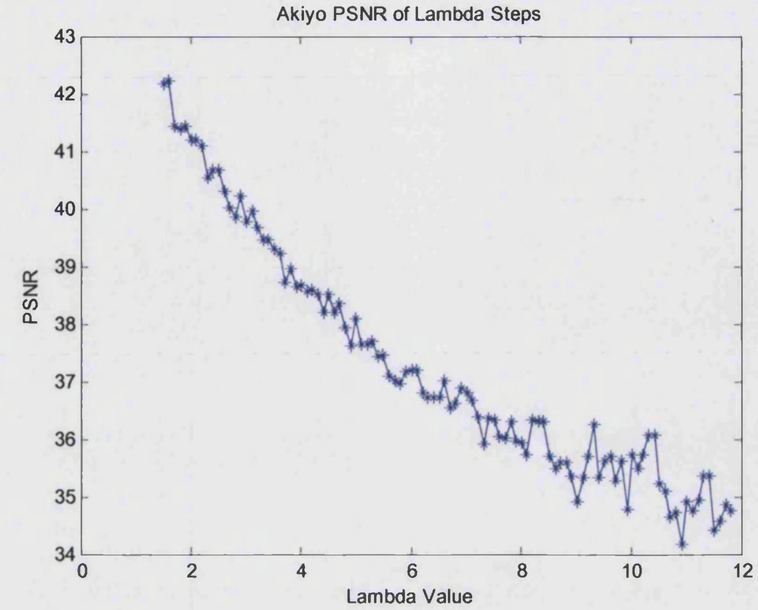


Figure 3.9 The PSNR-λ relationship for coding Akiyo

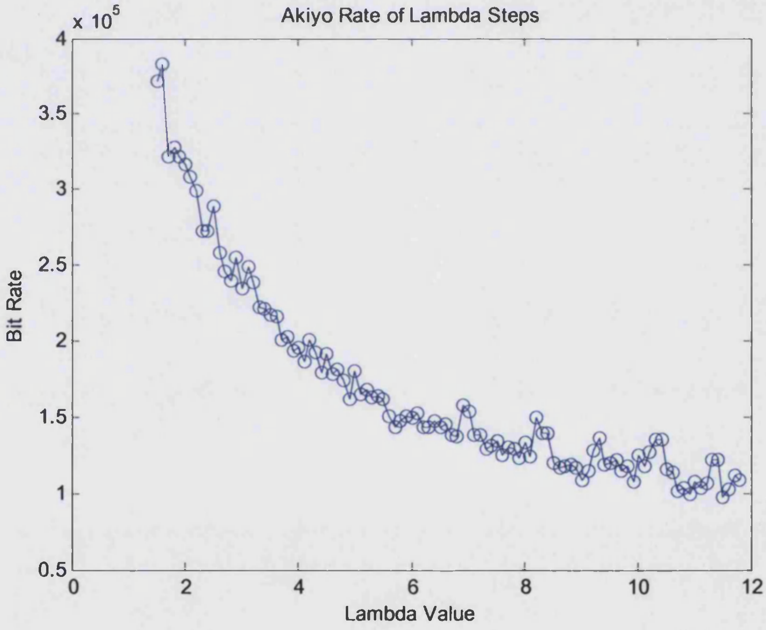


Figure 3.10 Rate- λ relationships for coding Akiyo

From Figure 3.9 and Figure 3.10, it can be seen that for coding Akiyo, along the decreasing λ value, the PSNR drops in a nearly linear manner. It goes down in nearly constant steps, while the resultant bit rate shows different changing trend. When λ varies between 1 and 4, the bit rate drops considerably. With higher λ values, changes over the rate drops with smaller steps.

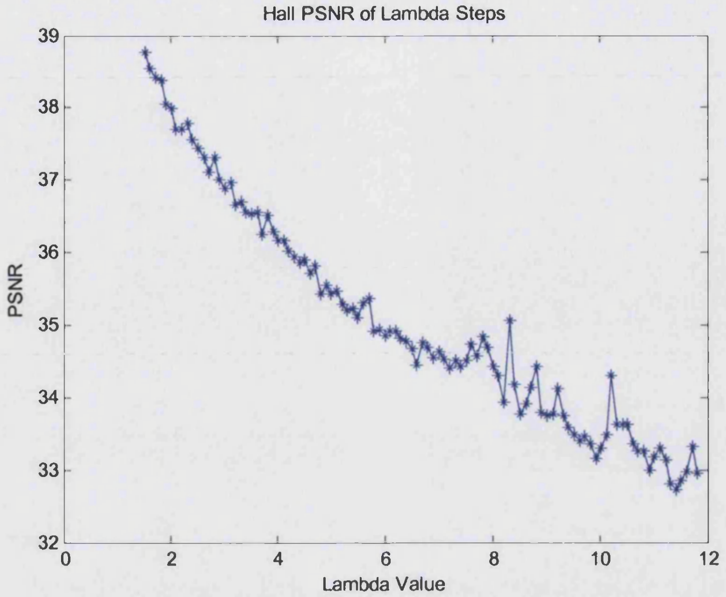


Figure 3.11 PSNR-λ relationship for coding Hall

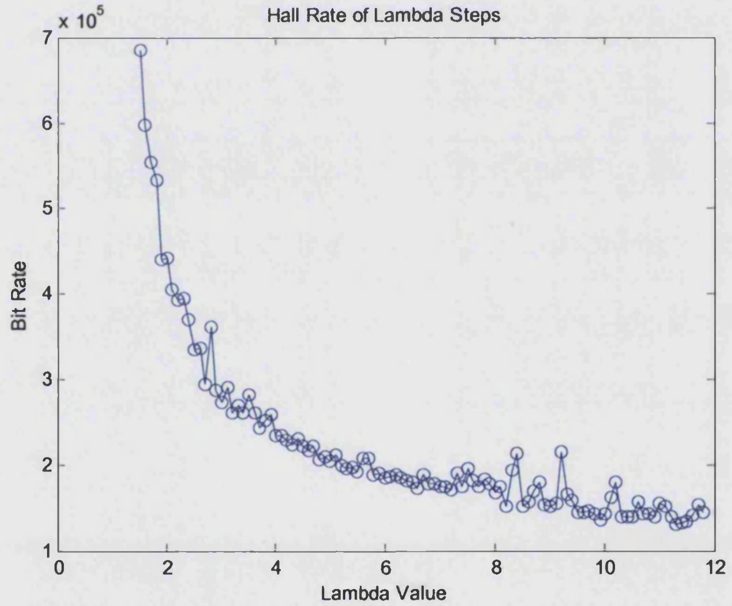


Figure 3.12 Rate- λ relationships for coding Hall

Figure 3.11 and Figure 3.12 show the situation for coding Hall. It is similar to the result for coding Akiyo. But in this case, significant drops of rate happens mainly when λ is smaller than 3.

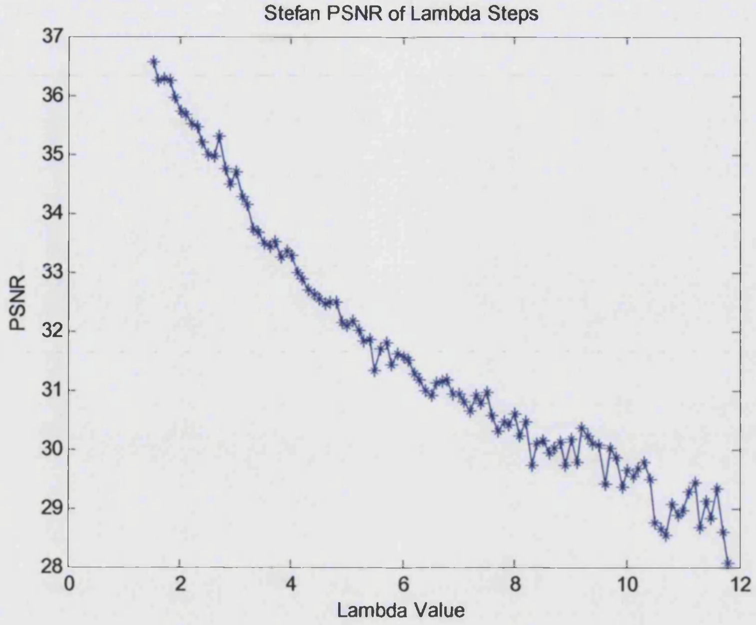


Figure 3.13 PSNR-λ relationship for coding Stefan

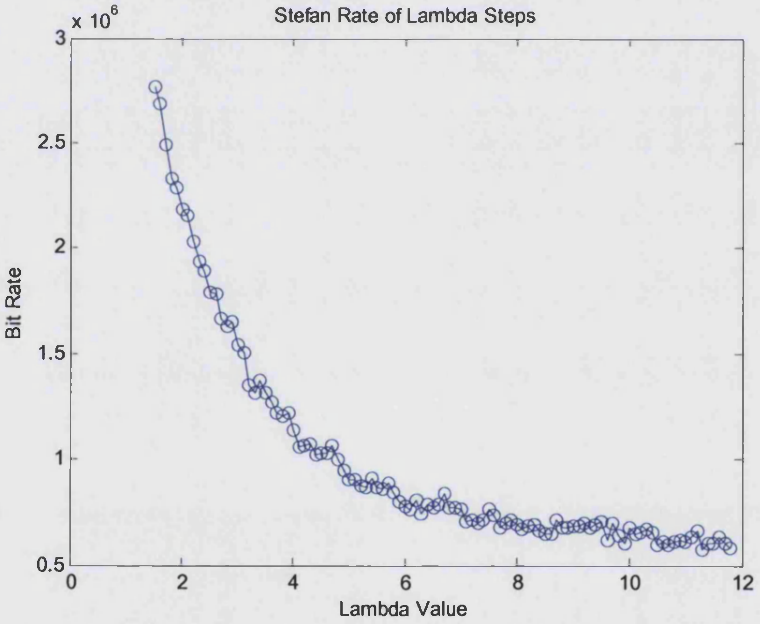


Figure 3.14 Rate- λ relationships for Stefan

Figure 3.13 and Figure 3.14presents the result for coding Stefan. Stefan has more motion content in it, and shows plots more similar to Hall. The significant drops of rate happen mainly when λ is smaller than 3, and PSNR shows a near linear relationship with λ changes.

3.5.3 Comparison with the JM9.2 rate control scheme

A bit rate control scheme is developed by Ma, et al [78, 79], and it has been adopted in JM 92 reference software. In the following, this scheme is used as a benchmark to compare with the proposed scheme.

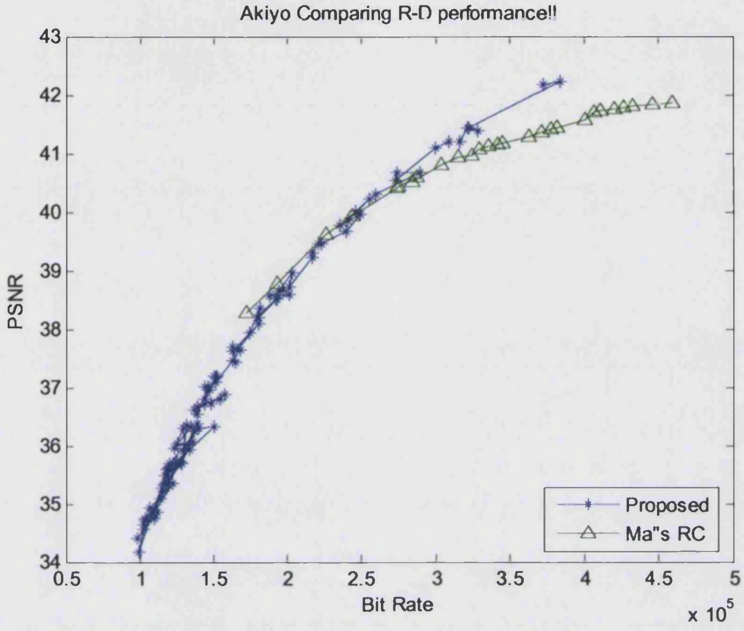


Figure 3.15 Comparison of Bit-PSNR result for coding Akiyo.

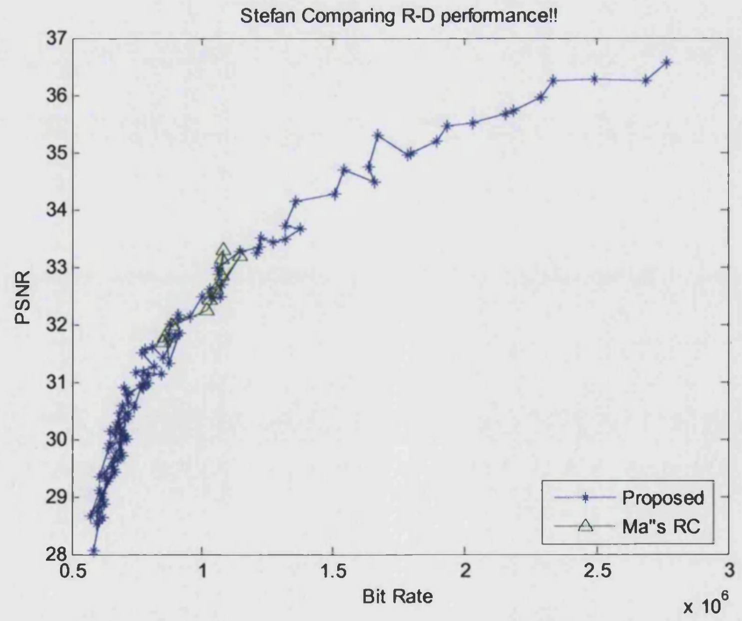


Figure 3.16 Comparison of Bit-PSNR result for coding Stefan

The results in Figure 3.15 and Figure 3.16show plots comparing Rate-Distortion performance of the proposed algorithm and the H.264 JM-92 rate control scheme [78, 79]. It can be seen that the proposed scheme gives acceptable performance. It generally provides a wide bit rate range and better PSNR for some medium and high bit rates. Especially for Stefan, the H.264 RC scheme is able to code only the sequence at very limited bit rates, and does not provide good average PSNR result.

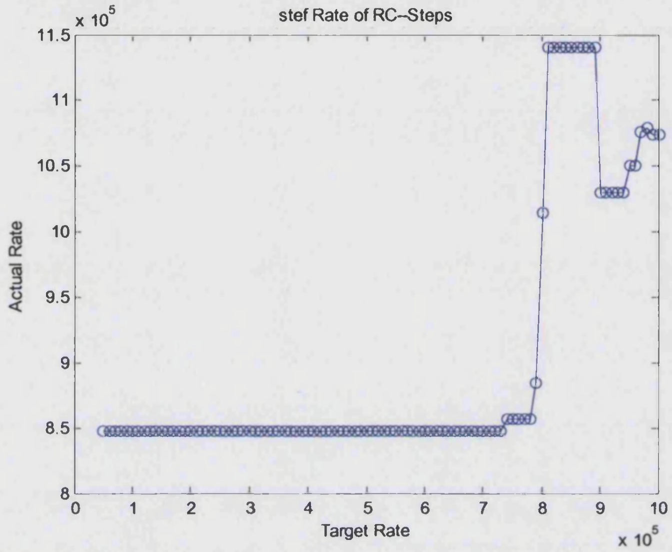


Figure 3.17 Actual BitRate vs Target BitRate of H.264 rate control for coding Stefan

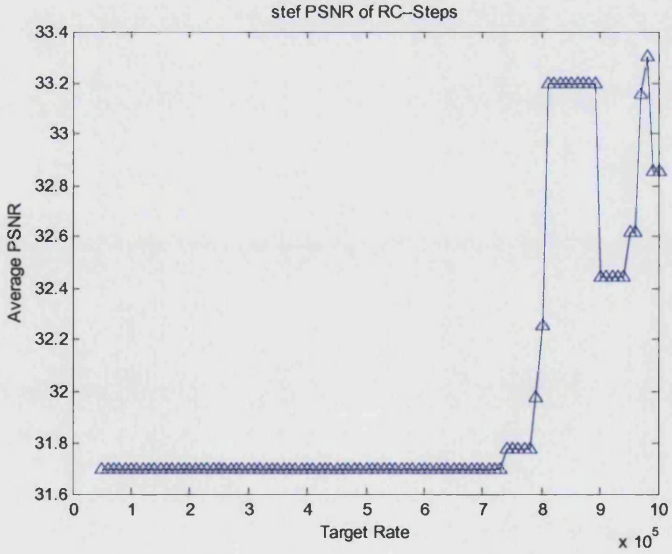


Figure 3.18 Resultant PSNR of H.264 RC for coding Stefan

Looking into the situation for coding Stefan, it can be seen that the RC scheme in H.264 has difficulty to achieve the accurate bit rate, as shown in Figure 3.18. It fails to meet the target rate and hence gives jumpy PSNR result.

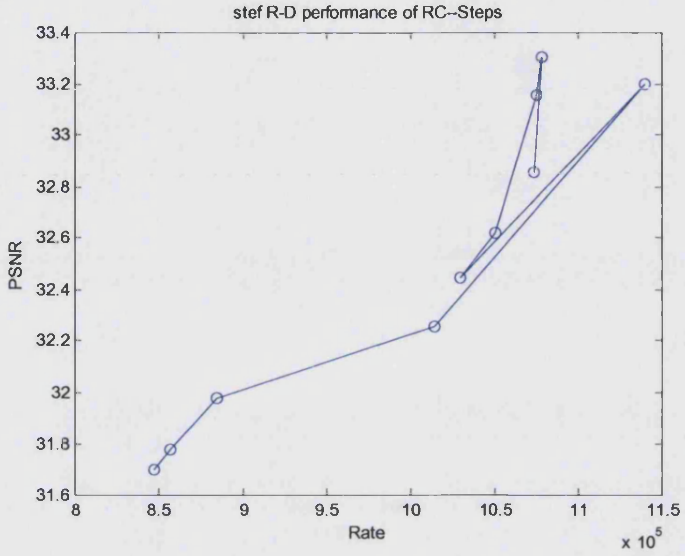


Figure 3.19 The Rate-PSNR results of coding Stefan by H.264 RC

The general R-D performance in this case is shown in Figure 3.19. The rate control scheme in H.264 does not manage the rate properly and it shows irregular plot in the R-D plane. The proposed algorithm shows obvious advantages in this case and this is discussed further.

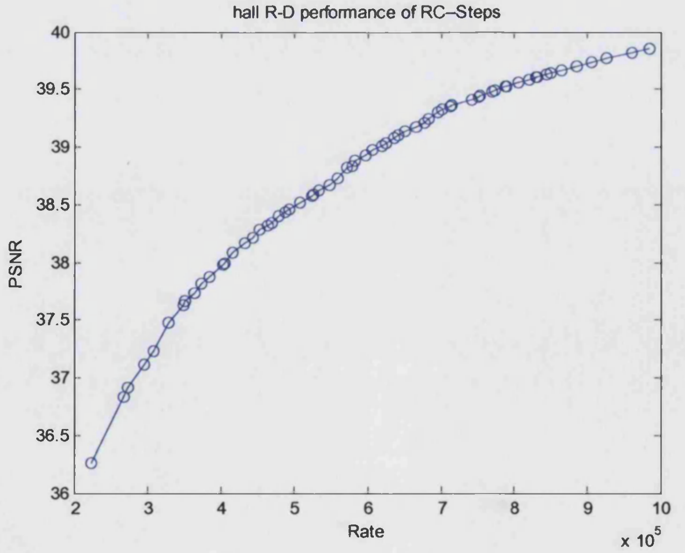


Figure 3.20 The Rate-Distortion performance of H.264 RC for coding Hall

The H.264 RC scheme shows a typical Rate-PSNR performance for coding Hall, which results in a curve with the convex facing upwards (see Figure 3.20). That is, at low bit rates, the image quality increases very quickly with the bit rate, and at high bit rate, the increase slows down.

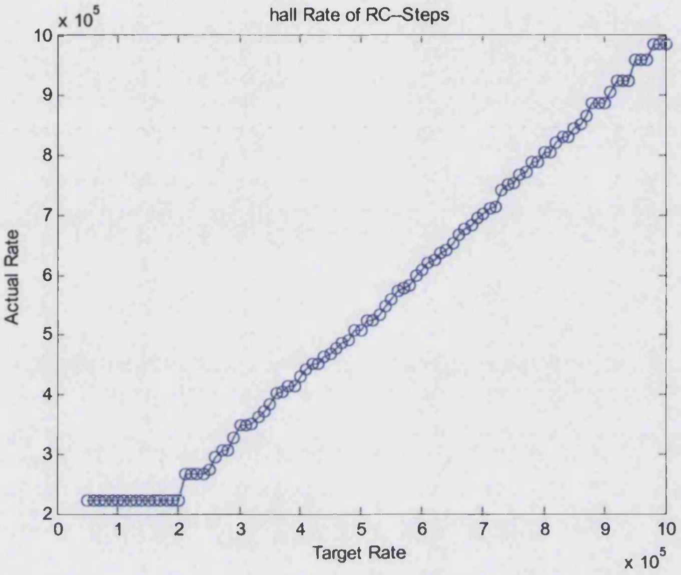


Figure 3.21 Actual Rate vs Target Rate of H.264 RC for coding Hall

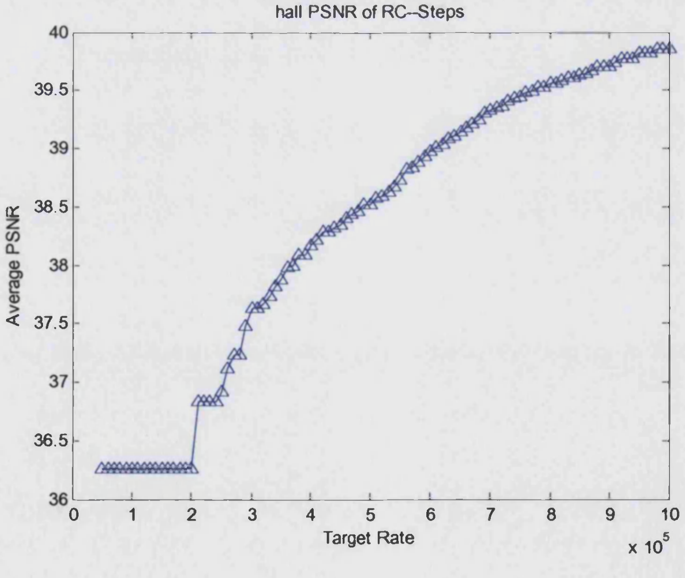


Figure 3.22 Resultant PSNR of different target rate by H.264 RC scheme

This is character is also illustrated in Figure 3.22. When coding Hall, the H.264 RC scheme gives very accurate control, as shown in Figure 3.21, which shows a near-linear Target-Result rate change trend. And the Target Rate-Actual PSNR relationship is very typical as well.

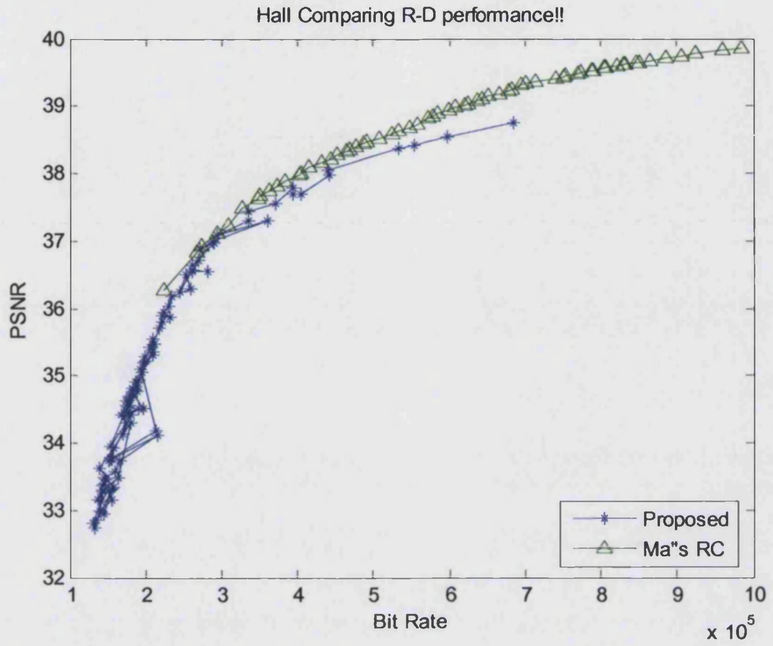


Figure 3.23 Comparison of Rate-PSNR result for coding Hall

In this case, the average PSNR of H.264 is very high and outperforms the proposed method (see Figure 3.23).

It should be pointed out that in this case the proposed method did not achieve very high bit rate due to the limited range of λ value that was set. When λ value is small, which corresponds to the high bit rate, it was not tested at fine enough intervals. If the λ value could be more finely tested, better performance at high bit rate can be expected.

3.6 Conclusion

In this chapter, a rate control scheme for H.264 video coding standard is proposed. Based on the constant slope concept, the R-D slope unifying scheme was proposed. The desire to improve the whole performance by making the R-D slopes of all parts close to the target slope value is there but is deferred to a later investigation. The monotonic property of R-D slope is not valid when applying at macroblock level. This makes the slope unifying scheme impractical.

The final scheme is based on Lagrangian multiplier. It enables an easier picking process. Experiments show that this scheme gives considerable gain for some sequences. The relationship between the bit rate and the parameter is also discussed. In the experiments carried out, Lambda values are chosen at regular intervals. It is found that at high bit rates, where the corresponding λ value is small, bit rate and PSNR vary with greater changes, in which case, finer sampling of λ value should be carried out.

Chapter 4 A HYBRID VIDEO CODER WITH H.264 AND MATCHING PURSUIT

4.1 Embedded coder and rate control in video coding

Accuracy and efficiency are two major difficulties with the conventional rate distortion control. JPEG is successful as an image compression standard, but has difficulty in controlling the coded file size. Due to the complexity and the unlimited range of the image content, it is almost impossible to precisely foresee the resultant file size or quality. Additionally, the quantization step options are limited which again affects accuracy. Therefore, it is quite often that the target bit rate can not be precisely achieved.

Accuracy and efficiency are both difficult for conventional coding schemes and it is especially hard to achieve them both simultaneously. The drawbacks come from the nature of conventional quantization, and can hardly be skipped. In the video coding standards, similar quantization schemes are exploited. Therefore, similar difficulties also exist. Embedded image coders are a new type of coding algorithms which allow precise and straightforward rate control as well as excellent coding performance.

4.1.1 Features of the embedded bit stream

The term "embedded" describes the feature of the bit stream produced by the encoders. It was first introduced in the well known EZW image coder by Shapiro [31]. In embedded coding, a signal (image) is coded at a bit rate Ri in such a way that the bit streams for all the other lower bit rates (R0 < R1 < ..... Ri) are progressively embedded within the bit stream for Ri and (D(R0) > D(R1) > ..... D(Ri)), where D(R) is the signal distortion at rate R [80]. This is depicted in Figure 4.1.

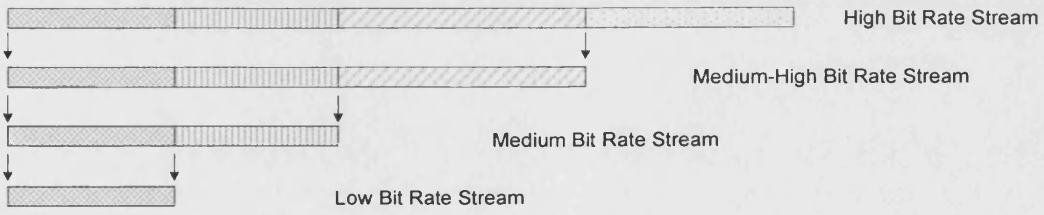


Figure 4.1 An example of embedded bit streams.

Such a feature comes from the special coding fashion. EZW [31] groups the same bits of each coefficient together into so called bit-planes, and treats them as whole units. The encoder scans from the most significant bit-plane to the least significant bit-plane. The higher the available bit budget, the more bit-planes can be covered in coding.

EZW is regarded as a milestone in image coding and it triggered a lot of interest in research on wavelet based image compression such as SPIHT, CREW etc [32, 33]. This trend was mainly driven by two features proposed by EZW: the potential performance improvement that might be achieved with digital wavelet transformation, as well as the idea of embedded output bit stream.

Embedded bit stream introduces a new feature in image coding which enables a much easier and accurate rate control method. Since bit stream of lower rate is embedded in front of the one of a higher rate, there is no need to decide the target rate before conducting the coding process. The encoder only needs to keep coding, until the criterion is met, that is, either the target rate is used up, or the target quality is achieved. The process is shown in Figure 4.2. As can be seen, repetition is avoided which assures that the whole coding process is greatly simplified.

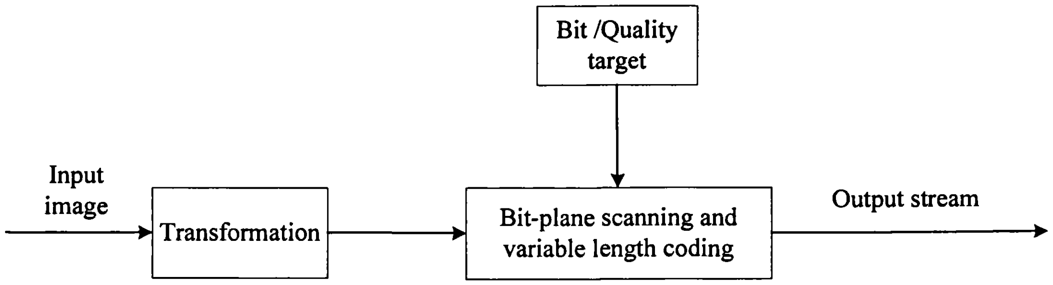


Figure 4.2 Rate control for an embedded image coder.

A straight forward and simple rate control can be easily achieved with embedded coding. Various bit rates can be implemented in a uniform coding fashion which saves the encoder of repetitive coding to achieve the target rate. The coding process is like applying successive approximation quantization to the coefficients, and therefore avoids the limit of quantization step options. Also, since the coder words allocated to the symbols are very small, the termination point is normally very close to the target size value. Hence, the rate target can be achieved very accurately.

4.1.2 Introducing embedded coding into video system

Most of the video coding standards [3, 15, 16, 43, 44, 47, 48, 50] and JPEG [38] try to control the bit rate by choosing different quantization parameters (QP) from the predefined table. Because of the huge variety of video content, it is very difficult to develop a direct relationship between the QP and the resultant bit rate. As described in Chapter 1, existing algorithms often use model to estimate the bit rate from the side information, such SAE. These models inevitably suffer from inaccuracy problems.

In contrast, the special feature of embedded stream easily avoids this difficulty and achieves the target bit rate with very small errors. Therefore, introducing embedded frame coders into video systems would offer a good solution for the rate control problem. In the following part, embedded coding methods are adopted into the video framework presented in Chapter2 to build a video system that is able to apply such a good rate control method.

4.1.3 Video coding with embedded frame coder

4.1.3.1 Candidate embedded frame coders

The intra frames in a video sequences is treated as normal still image. So the main difference between video frame coding and still image coding lies in the coding of displacement frame difference (DFD) frame, or residue frame coding. This part of work focuses on coding the DFD frame with embedded coder.

The previously presented embedded coding algorithms are all based on DWT transformation. The multi resolution nature of DWT makes a lot of other features possible, such as the SNR scalability and spatial scalability. But regarding the compression efficiency, the high coding performance of the schemes might come from a good data structure rather than an inherent nature brought by DWT[81]. By properly organizing DCT coefficients, the coding performance might also be improved [82].

A lot of effort has been made to develop embedded coding upon DCT coefficients [81, 83, 84]. Poh and Monro [83] proposed a DCT based embedded coding scheme specially designed for motion displacement frames coding. This is the first coder that is going to be used here.

In this method, the residual frame is first divided into 16x16 macroblocks following the motion prediction procedure and then according to a bases selection method the blocks are transformed by 8x8 or 4x4 DCT into the frequency domain. Those resultant coefficients are then coded in a bit plane-wise manner. A significance block map is introduced to arrange those coefficients. This map is a bi-level structure to help coding the zeros jointly. A bit plane is converted into significance block map M and significance sequence S. Both M and S are rearranged into a long series of bits and then entropy coded with a Golomb code [85, 86] .

For purpose of comparison, another scheme to be used is JPEG2000 [49]. The lossy coding scheme of JPEG2000 is based on embedded block coding with optimised truncation (EBCOT)[54]. Like many embedded image coders, EBCOT is also a wavelet-based coding algorithm. It has proven to be very effective in compressing still images, and here it is tested against DFD frames.

4.1.3.2 System specification and experiments

For an inter frame, the encoding is accomplished in a two pass procedure. In the first pass, the system performs the motion estimation, including the mode and reference frame selection. H.264 divides a frame into 16x16 macroblocks, and the following motion estimation and compensation is performed at MB level. Both the chosen schemes process the whole residual frame rather than small macroblocks. Therefore, in this pass, the video framework collects each predicted MB individually and generates a displacement frame after the motion estimation is finished. All the control data from the first pass are saved and encoded. In the second pass, the saved displacement frame is encoded by the adopted embedded coder.

It should be noted that in a conventional macroblock level inter frame coding process, an MB is allowed to be coded as either inter or intra mode. In H.264, an intra prediction mode is introduced which exploits the correlation among adjacent MBs. Such prediction is made basing on the information from the already coded MBs. When the motion residual coding is performed upon the whole image, such information is not available and hence prediction is impossible. Therefore, we disable this mode during the encoding.

The embedded frame coding feature makes it very convenient to implement the rate control. It is introduced with the advantage of employing embedded coding in straightforward rate control. In this framework, we enable the inherent RC function of

H.264 reference code. For each frame, in the first pass, the bits number used on displacement is recorded, and fed to the proposed embedded coder in the second pass.

Finally, the encoded inter frame is reconstructed by the decoder of the proposed algorithm and passed back for the usage of predicting future frames.

Experiments are carried out using this two pass H.264 based video coder which is introduced in chapter 3. The two embedded coding algorithms presented above have been integrated into the framework respectively to perform the inter frame coding.

The performance comparison was carried out on the first 100 frames in four common test video sequences: Akiyo, Foreman, Stefan, and Hall, all in CIF (352x288) resolution. Coding is performed on luminance component only. Since the replaced part is applied on inter frames, an extreme frame structure I-PPPPP... is applied. An I frame is followed by 99 P frames.

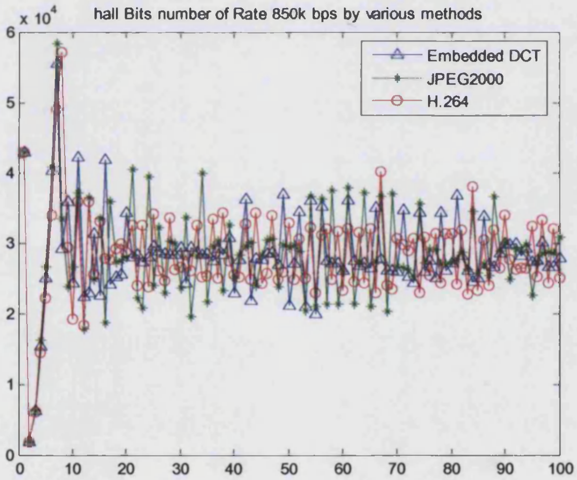


Figure 4.3 The bits of each frame for coding Hall at 850 kbps

Figure 4.3 shows the bits number used for each frame of Hall when coded at 850 kbps by various algorithms. It is clearly shown that the two embedded algorithms well follow the bit rate applied by H.264, which indicates their excellent potential advantage for rate control. The small variation is mainly due to the different coding efficiency which results in different distortion in the following frames.

However, in Figure 4.4, it can be seen that the H.264 easily outperforms the two candidate coders and always takes the top position of the three plots.

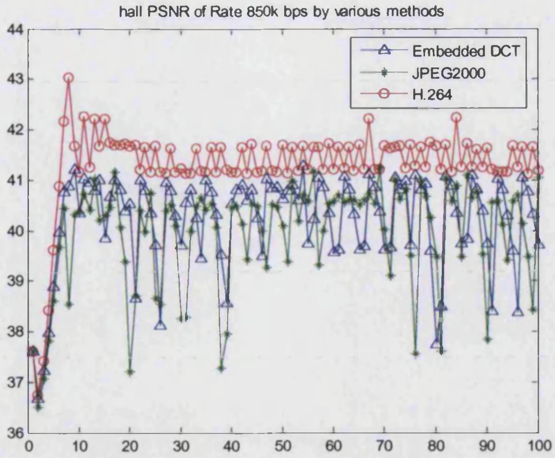


Figure 4.4 The PSNR of each frame for coding Hall at 850 kbps

An over all result of coding four sequences are shown in Figure 4.5.

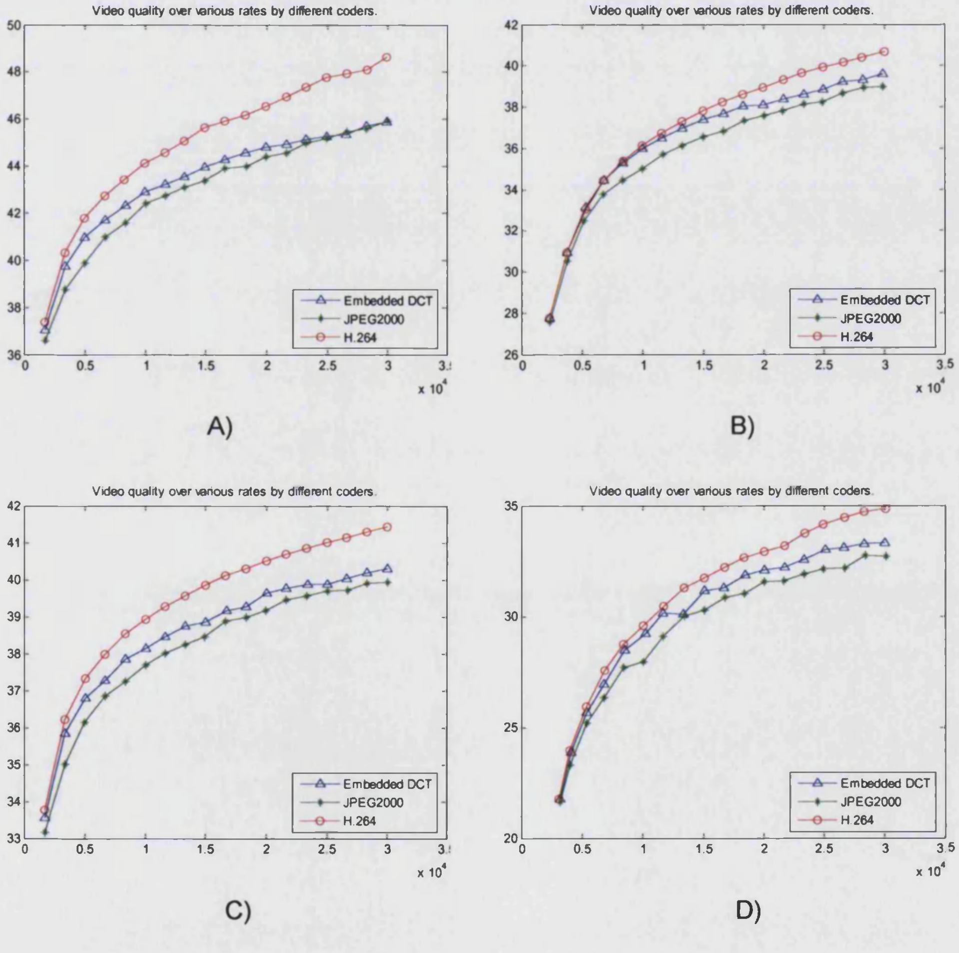


Figure 4.5 The overall Rate-Distortion results of coding four sequences with various schemes, A). Akiyo B) Foreman C) Hall D) Stefan

4.1.3.3 Conclusion

From Figure 4.5, it is clearly shown that the two embedded coding schemes show very good ability to implement the target rate. They always meet the target rate with very little deviation. However, H.264 original coding provides an outstanding coding efficiency and outperforms the other two schemes over most bit rates. Also, these two algorithms provide pretty good coding efficiency at low bit rate, especially when coding Foreman and Stefan, where more motion content is involved. To achieve comparative coding performance to H.264, new algorithm has to be exploited.

4.2 Matching pursuit, an alternative to transform coding

Although embedded coding helps to achieve accurate and straightforward rate control, the techniques in literature do not provide as high coding efficiency as H.264 does. A more powerful encoder is therefore required. As an alternative to conventional transform coding, matching pursuit (MP) coding enables outstanding coding efficiency, as well as precise rate control.

4.2.1 Review on matching pursuit algorithm

Matching pursuits (MP) [87] is a signal processing technique that has stimulated much recent research. In particular, it has proved to be a powerful alternative tool to transform coding for visual signal compression[88]. The iterative nature of the MP process also has much potential for the development of simple and accurate rate control schemes for video compression.

Matching pursuit was introduced as a signal processing algorithm by Mallat and Zhang [87]. It is a recursive process that iteratively decomposes a signal using basis functions selected from a predefined code book. The signal is then represented as a weighted combination of the chosen.

The idea of matching pursuits is to decompose a signal into a linear expansion of waveforms, which belong to a redundant dictionary of functions $D = \{\varphi_\gamma\}$, with $\|\varphi_\gamma\| = 1$ for all γ. MP is a recursive procedure in which, at each time of iteration, a function $\varphi_{\gamma n} \in D$ that best approximates part of the signal is chosen. So, after *m* iterations MP decomposes the original signal f into a sum of dictionary elements, so that

$$f = \sum_{n=0}^{m-1} \alpha_n \varphi_{\gamma n} + R^m f \, , \tag{4.1}$$

where a_n is the matching pursuit coefficient (or inner product) given by

$$a_n = \left\langle \varphi_{\gamma n}, R^n f \right\rangle \tag{4.2}$$

and $R^m f$ is the mth order residual vector after approximating the signal in the direction of $\varphi_{\gamma n-1}$.

It is often impossible to obtain an exact reconstruction of the signal. Firstly, infinite number of iterations cannot be completed, and secondly, the matching pursuit coefficients are quantized to $\alpha_n ' = Q(\alpha_n)$ prior to the calculation of the residual $R^{n+1} f$. Considering these two factors, the reconstruction of the signal is given by

$$\hat{f} = \sum_{n=0}^{p-1} \alpha_n ' \varphi_{\gamma n} \tag{4.3}$$

In general, MP algorithms can be applied to any set of redundant basis functions. Mallat and Zhang [87] proposed to expand a signal using an over complete set of Gabor functions.

4.2.2 The application of MP in video coding

In hybrid video coding systems, motion estimation and compensation models are applied to video sequences to reduce their temporal redundancy, producing motion residual frames. Since a large part of the temporal correlation has been removed, motion residual frames generally exhibit a very sparse data structure, which is very well suited for encoding with MP algorithms. In [88], Neff and Zakhor introduced MP into video coding systems. In particular, they used MP to replace DCT encoder for motion residual frames in an H.263 coding system. The new MP video coder was shown to be very effective, producing a significant increase in the PSNR performance over the H.263 system.

The widely used hybrid video encoder employs motion compensation to predict the current frame from previously reconstructed frames. The predicted image is then subtracted from the current original image to produce the motion residual image. Since most of the correlation has been removed, a residual image often contains some sparse and clustered data structures. The mechanism of MP coder is very efficient for encoding sources with sparse data distribution structure. Neff and Zakor first proposed the incorporation of MP coder into a hybrid video coding system [88]. This approach combines the advanced motion models of hybrid video systems with the high efficiency of MP coder and provides very good coding performance. In their experiments, the MP coding scheme is adopted into the H.263 system, and outperforms H.263 with a substantial efficiency improvement.

4.3 The Bath University Matching Pursuit project

The Bath University Matching Pursuit (BUMP) is a project that develops visual signal compression schemes with MP algorithm. BUMP improves the coding performance by introducing some important features into MP, such as wavelet pre-transformation, MERGE algorithm for encoding atoms [89], a quick atom searching method [90], new codebooks based on an sequential basis picking algorithm [91], etc. Also, it extends the MP algorithms to still image coding [89].

The Bath University Matching Pursuit project (BUMP) further develops the idea of MP coder. It aims to developing efficient matching pursuit algorithms for encoding video signals. For the purpose of improved compression efficiency, BUMP introduces many of new features, including:

1) Designing new codebooks. MP algorithms try to decompose a signal by transmitting the approximation between the signal and the basis functions. Therefore, proper chosen bases are vital for providing a high coding efficiency. In BUMP, a special basis picking scheme is exploited to construct an effective codebook [91]. The codebook not only holds basis functions that have the ability of conveying the image information with high fidelity, but also has a very small size, which in turn needs fewer bits to encode the basis identity than a larger codebook requires.

2) Introducing wavelet pre-transformation. The nature of MP makes it very effective at encoding data of sparse and clustered structures, such as residual frames. In our previous work [89, 92], it has been shown that by applying wavelet transformation as pre-processing stage, the energy can be further concentrated so that the subsequent atom searching and coding processes can be more efficient.

3) Extending the MP coder into still image encoding [89]. From the same motivation as described above, we apply wavelet transformation to still images so the energy is more

focused and the data becomes sparse and locally clustered, and thus more suitable for MP encoding.

4) Separable atom searching strategies. Various atom searching schemes are studied [90]. Some suboptimal schemes allow a quicker searching procedure. The trade-off between the performance and the complexity is studied and the best compromise is exploited.

5) Precision limited quantization (PLQ) for quantizing the atoms' amplitude. PLQ was first introduced for wavelet based image in order to design a visual lossless quantization method according to psycho-visual experiments [93]. Later it was found to be effective in helping to improve the compression performance of MP video coders [94]. PLQ provides an effective alternative quantization scheme for MP.

6) Multi-pass Embedded Residual Group Encoding (MERGE) coder for atom encoding. The MERGE coder classifies the atoms into groups according to their quantized amplitude and transmits them by encoding their positions. Together with PLQ, the BUMP coder is able to provide a perfectly embedded bit stream, and thus potentially allows a more accurate rate control for the video coder [92].

With all the above features, BUMP is able to provide MP visual compression algorithms of low complexity and high efficiency for various sources.

Neff and Zakhor's success in applying MP to video coding triggered much interest in MP based video coding schemes. Meanwhile, tremendous advances in conventional hybrid video coding have been made, many of which are incorporated in the new video coding standard, JVT/H.264 [25, 26]. H.264 is the state-of-the-art video coding standard and outperforms all its predecessors in sense of coding efficiency [26, 27, 50, 71]. Therefore, the incorporation of MP within an H.264 framework has much potential for improved MP-based video coding.

In [92], the above introduced BUMP coding algorithm is applied into video coding, and a hybrid video coding system based on H.264 with matching pursuit was proposed, the structure shown in Figure**.

The state-of-the-art H.264 standard represents the latest advances in video coding and outperforms most video coding schemes presented in the literature. One of the features of H.264 that contributes to such a high efficiency is the advanced motion estimation and compensation model. In our proposed video coder, all the advanced features of the motion model are preserved, including the multi block shapes, higher motion vector accuracy and multi reference frames.

The encoding process of a frame is carried out in macroblocks (MB) in H.264. In the proposed system from [92], each MB is collected and rearranged into a displace frame difference (DFD) image, which is then fed into the MP coder. The proposed MP coder first performs a two-scale wavelet transforms over the DFD frame, and then applies the recursive atom finding and repairing procedure using our distinctive code book. Finally, the atoms are quantized with a PLQ scheme and encoded with the MERGE coder to produce a fully embedded bit stream.

4.4 A hybrid video coder with H.264 and matching pursuit

In this section, a novel video coding system is presented that is based on matching pursuit and H.264 [92]. The motion compensation and estimation model of H.264 is employed in this system, and the motion displacement frames are encoded by the matching pursuit scheme developed in BUMP. The combination of these two techniques retains the advantages from both parts, and produces a highly embedded data stream that can be easily truncated at any desired point, which allows accurate rate control.

The basic structure of matching pursuit algorithm developed in BUMP is as introduced in earlier sections. In this work, the MP coder incorporates a wavelet pre-transformation to 2 scales, and a high performance 8×8 2d separable Gabor dictionary is exploited to find a series of atoms which are then coded by the MERGE coder. In what follows, the design of the coder is discussed in more details.

4.4.1 The motion estimation and compensation—retained from H.264

In general, this video coding system is built on the same video framework as presented in earlier chapters. Similar to the experiments in chapter 4, the motion estimation and compensation part of the video coder is inherited from H.264, and retains all of its advantages. The motion module provides excellent performance that comes from a few new features introduced in H.264, including various motion prediction shapes, higher motion precision, larger reference frame memory, etc. We keep this module to produce both motion vectors and DFD frames. Yet some of the features are for H.264 only and cannot be utilized.

In H.264, for each macroblock in the DFD frame, there are a few candidate coding modes (Intra, 16x16, 8x16, 16x8, 8x8, 4x8, 8x4, and 4x4) [25, 69, 95]. To make its Intra coding more efficient, H.264 employs so called Intra Prediction scheme, which uses the available reconstruction pixel values from neighbouring blocks to predict the pixels in the present block. This scheme is specially designed for H.264, in which a frame is encoded macroblock by macroblock. But in MP coder, each residual frame is treated as a whole unit, and the location where an atom is found earlier would be coded earlier. There is no way how these atom locations could be known before encoding is carried out. In this case, such intra prediction is not possible. To prevent this conflict, the intra prediction mode is switched off in our coder.

Apart from this, most advantages of the motion module are reserved, including the variable block sizes, quarter pixel motion vectors accuracy, usage of multi-reference frames, etc.

4.4.2 The system structure and bit rate calculation

A unified video coder based on H.264 with matching pursuits was proposed in [92]. The structure of this video coder is shown in Figure 4.6. The proposed MP encoder for DFD is specified in the dashed rectangle. As can be seen in Figure 4.6, the original frame is compared to the previously reconstructed frames to produce motion vectors and the DFD frame. The latter is taken as input of the MP coder, and the reconstruction is built with the MP decoder, and passed back for purpose of future motion estimation.

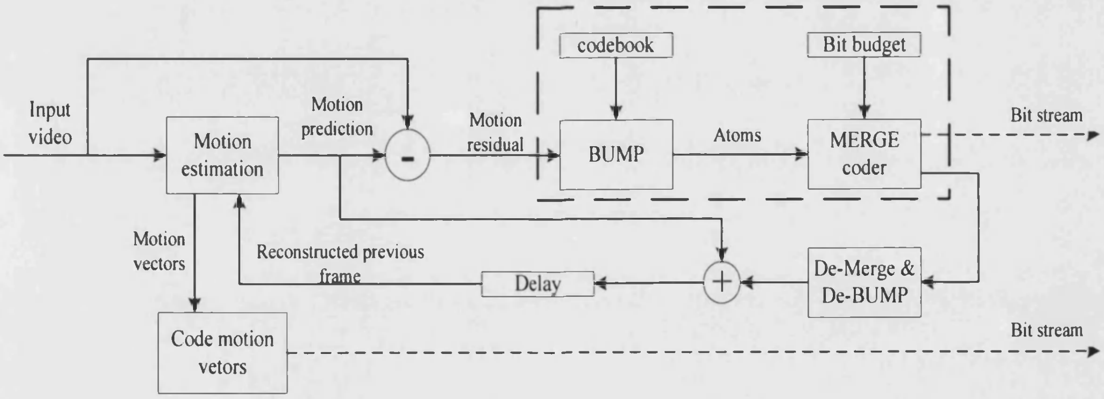


Figure 4.6 The proposed hybrid video coding system

The proposed DFD module is a hybrid frame encoder in the sense that it exploits the wavelet transform as pre-processing functionality, and followed by a MP module to carry the coding.

4.4.3 Bit number calculation and some special consideration

For the purpose of comparison, the rate control mechanism in H.264 is enabled, and set at frame level, i.e., Basicunit=396 (MBs) for CIF. Although the bit stream created by MERGE coder is of a size very close to the desired one, there are still some minor mismatches. For the purpose of accurate rate control, the actual number of bits used needs to be passed back to the framework.

The number of bits allocated for each frame is recorded and the actual bit budget for the proposed BUMP coder can be calculated by subtracting the number of unavailable bits from this recorded number, as:

$$Budget = Recorded\_bits - Unavailable\_bits; \tag{4.4}$$

where the $Unavailable\_bits$ includes those for coding motion vectors, block modes, etc.

In practice, the MP coder first searches to find overabundant atoms. Then, after obtaining the actual desired bit budget, the MERGE coder can be instructed to choose a proper part of them to code. The number of actual bit usage can be feed back to the code using the interfaces of the framework as described in Chapter 3.

4.5 An ad hoc rate smoothing scheme for the proposed video coder

4.5.1 A glance at the R-A character of BUMP coder

In general, MP algorithm is a recursive algorithm which at each recursion tries to search and deliver the atom that is most worth of encoding, in another word, reduces most distortion. And therefore it can be expected that the amplitude of the found atoms decreases as the encoding process goes on.

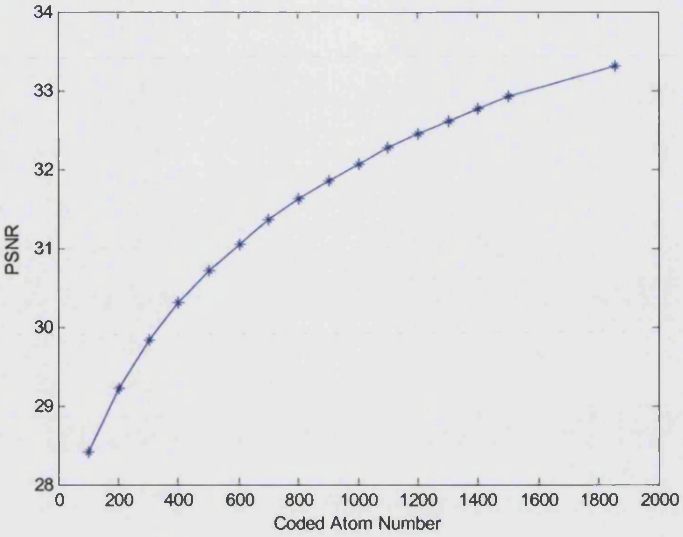


Figure 4.7 Atom number vs PSNR for the 5th frame of Stefan.

Figure 4.7 shows an example of how a DFD frame's PSNR increases as more and more atoms are delivered. The slope is steep in the beginning, and turns flat later on. This can be expressed mathematical form as following,

Define the PSNR slope as $\lambda = \frac{\Delta PSNR}{\Delta Atom}$, then $\lambda \downarrow$ when $Atom \uparrow$.

Obviously, the atoms found in the earlier stage contribute much more to the PSNR increase than the later ones do. Experiments show that such a slope changing trend is a common phenomenon when applying MP based video coding over different video sequences, so this character should be considered in bit allocation. A good bit allocation should ensure more of such early atoms are encoded.

4.5.2 A rate allocation scheme with atom number limitation

BUMP coder produces an embedded output bit stream that facilitates easy implementation of rate control. A straightforward way to code a frame at desired bits number is to find far redundant atoms in the stage of BUMP, and code part of them in stage of MERGE, atom number depending on available bit budget.

Such rate control scheme is very similar to the one used in the system in Chapter 4. Actually it follows the bit allocation guidance from the H.264 rate control scheme, which might not be optimal in the sense of PSNR. In fact, the process of coding atoms can be controlled according to other parameters.

First, the rate-distortion performance of the proposed MP coder is looked at and a simple experiment is carried out. Stefan is encoded with the proposed video coder, where the all motion residual frames are encoded by the MP coder. The frame rate is set lower so that the 5th frame is predicted by the first frame. Let the coder to encode the residual frame at various target atom numbers. The plot showing PSNR changes along with the atom number increase can now be obtained as shown in Figure 4.7.

It can be clearly seen that the atoms that are encoded in earlier stage help to achieve considerable PSNR increase, while the coding gain brought by later atoms are much smaller. Therefore, it is important to give all frames some bits to code the first few atoms that can be found with the MP coder. In contrast using too many bits on one single frame does not provide as much coding gain. Therefore, smoother bit allocation among frames should be considered.

Another parameter for guiding the coding process, MaxAtom is introduced now. MaxAtom is the allowed maximum number of atoms to be coded in each frame. Now, each time the MP coder encodes a DFD frame, it must consider two criteria, the encoder stops coding when either the available bit budget is used up or the number of coded atoms number has reached the value of MaxAtom.

Setting MaxAtom prevents the limited total budget of bits from being allocated too heavily on a few frames which might cause bits insufficiency for other frames. According to the rate-distortion function, for each frame, the first few bits provide the most drastic distortion reduction. So, giving a smooth bits allocation along the frames would help to use bits in such a way that the PSNR could be increased most, and therefore improve the overall PSNR result.

For the same reason, the initial quantization also affects the coding results. Setting a proper initial quantization parameter would enable to code the front frames at similar quality to those frames in the back of the sequence.

Our experiments show that the MaxAtom and initial quantization parameter affect the average PSNR and bit rate. Hence, a simple optimization process can be carried out by properly setting those parameters.

4.5.3 Experiments

4.5.3.1 Specifications

Since this work focuses on inter frame coding the I frame coder of H.264 is used, and all inter frames are coded with the proposed BUMP and MERGE coder. An IPPP... frame structure is applied for coding the sequence and total 100 frames are coded each time, i.e., one Intra frame and 99 Predictive frames. Experiments have been carried out using Stefan and Hall, both of CIF size (352x288). The H.264 uses CABAC option for entropy coding. For simplicity, all experiments are carried out on luminance components only.

In proposed MP coder, an small 8x8 size codebook is employed [91]. The codebook is developed from 8 separable basis functions shown in Table 2.1. This design greatly simplifies the complexity of the dictionary components coding so that each basis can be transmitted by two index numbers within the range of [1, 8]. For the PLQ in MERGE coder, the value of PL is set to be 2.

8
1 5 7 9 7 3 3 3 7 7 7 5 7 5
1.0000
0.4027 0.4666 0.4901 0.4666 0.4027
0.0013 -0.0668 0.2918 0.6400 -0.7045 0.0668 -0.0005
0.3165 0.3288 0.3379 0.3435 0.3453 0.3435 0.3379 0.3288 0.3165
0.0000 0.0363 0.5412 0.8393 0.0000 0.0363 -0.0010
-0.3832 0.8404 -0.3832

Table 4.1 The 8 bases functions used in the experiments

And 2 scale DWT transformation is used for pre-processing to the motion displacement frames. The famous 9/7 wavelet filter pairs [58] is applied.

Coefficients	Lowpass	Highpass
0	+0.602949	+1.115087
±1	+0.266864	-0.591272
±2	-0.078223	-0.057544
±3	-0.016864	+0.091272
±4	+0.026729	

Table 4.2 Analysis low-pass and high-pass filter bank

The proposed hybrid video coder is used to compress Stefan and Hall at various bit rates using different parameter values. To find out how the two parameters would affect the result, different settings are tried. The following plots show the average rate and PSNR results. Each point represents one MaxAtom-Initial QP combination. For every single bit rate, the points are presented in two ways. First, they are grouped according to the MaxAtom value, and in the second, they are grouped by the initial QP applied.

The best setting of those parameters can then be found for all bit rates in experiments and good results can be achieved with the proposed video coder.

4.5.3.2 Results and discussion

It can be seen that the ultimate performance can be improved by changing the value of MaxAtom, as well as the Initial QP. If the MaxAtom is set to be too small a value, there might not be enough atoms coded, and the bit budget is not fully utilized, hence the quality is poor. Yet the experiments show that the idea of setting a far redundant higher bound of MaxAtom does not always yield the best results either. Table 4.3 shows the best MaxAtoms obtained in experiments. The optimal value of MaxAtom depends on the sequence as well as the target bit rate. It generally increases with the average bit rate, and drops back to a low value at high bit rate.

Target Bit Rate	Best MaxAtom for Stefan	Best MaxAtom for Hall
100k bits/second	1000,2000,3000 and 4000	1000,2000,3000 and 4000
150k bits/second	1000	800, 1000
250k bits/second	1500	2000
350k bits/second	1500	3000
500k bits/second	800	800
750k bits/second	2000	N/A

Table 4.3 Best MaxAtom values at various bit rates

Table 4.3 shows that when coded at 100kbps, MaxAtom of 1000-4000 give identical results in both sequences. In case of low bit rates like this, the actually coded atoms are always far fewer than the higher bound, so setting MaxAtom to 1000 or 4000 makes no difference. In Figure 4, this is shown by the clustered points at the up left corner. In contrast to the above, Initial QP has a pronounced more effect on the final PSNR.

Situations change at the rate of 150kbps, and MaxAtom of 800 and 1000 show only slight advantages. This might arise because of two reasons. First, the rate control scheme accumulates information for adjusting the bits along the coding process and thus a few early frames' bits cannot be properly allocated. If they take too many bits to code, the encoder would have to reduce the bits for later frame to achieve the target bit rate and this is what causes the temporal imbalance in bits allocation. A smaller MaxAtom value helps to control such disparity allowing more bits to be used for coding later frames in the sequence. Second, because of the mechanism of Matching Pursuit, the few atoms found in the early stage often contribute to dramatic distortion reduction. So, reserving a few bits for later frames enables more "early atoms" to be encoded, and thus helps to improve the average PSNR.

This effect is shown in Figure 4.8, where Stefan CIF coded at 150k is taken as an example. Clearly from the figures show the PSNR and rate changes of the whole sequence. MaxAtom 800 helps the bit rate to converge more quickly and thus the PSNR of later frame is higher, and as a result the average PSNR is higher.

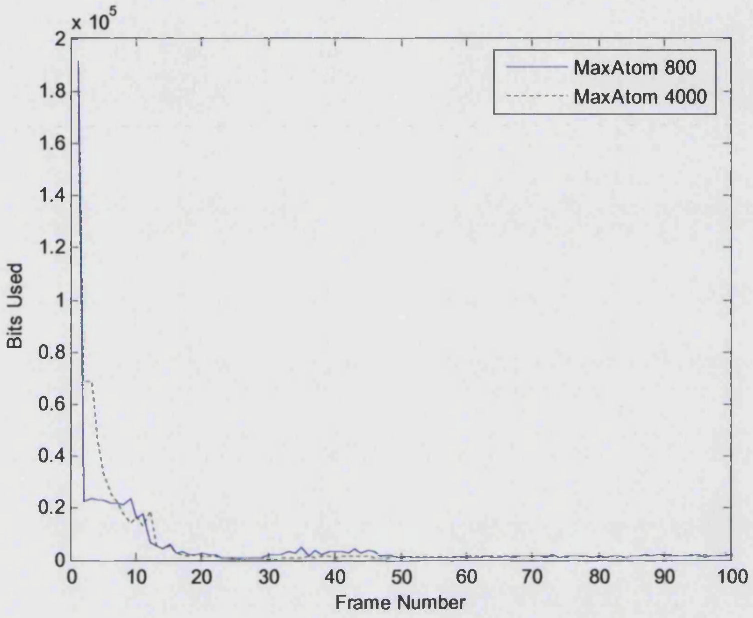


Figure 4.8 Bits of coding Stefan by various MaxAtom values

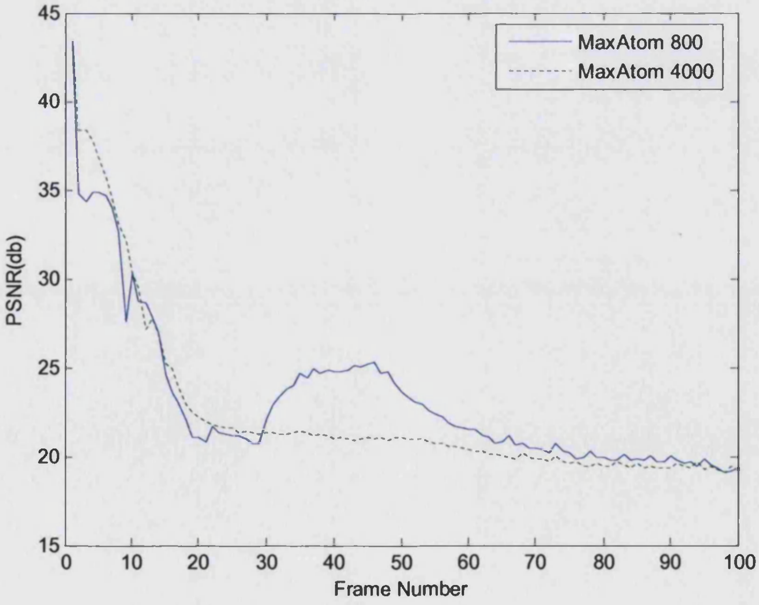


Figure 4.9 The frame PSNR results of coding Stefan with various MaxAtom values

The parameter setting search for different target bit rates is done to get the performance plot, which is shown in Figure 4.10 and Figure 4.11. The plots show the performance of H.264, H.263, and the Neff & Zakhor MP video coder.

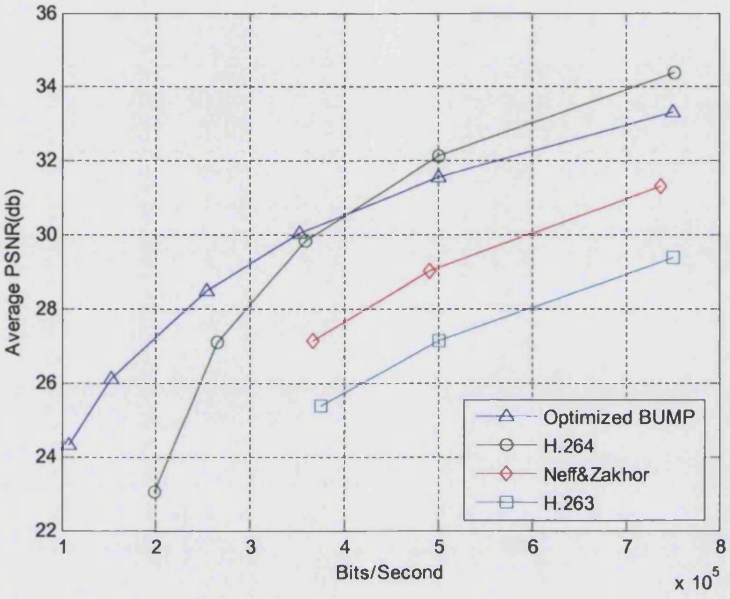


Figure 4.10 Rate/Distortion comparison on Stefan CIF

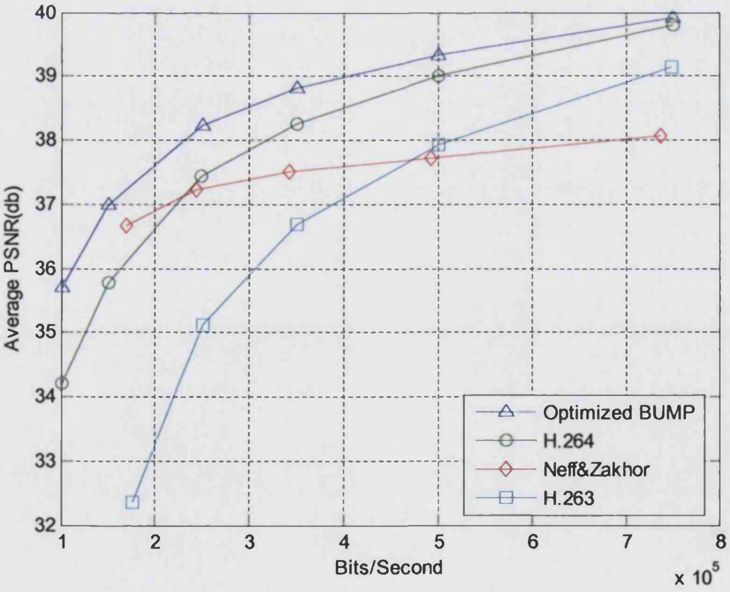


Figure 4.11 Rate/Distortion comparison on Hall CIF

It is seen that the tuned BUMP video coder is competitive to H.264, and outperforms H.264 at medium and low bit rates. And, also benefiting from the advanced motion module, it greatly outperforms the Neff & Zakhor's H.263 based matching pursuit video coder.

4.6 Conclusion

A very efficient coding scheme with hybrid MP frame compression is presented. The highly scalable bit stream produced by the BUMP schemes reserves the embedded coders' advantage in rate control, and also provides an outstanding coding efficiency. Considering the rate and distortion character of the MP coder, a simple ad hoc bit allocation scheme is proposed. Experiments show that by properly setting the two parameters, this video coder provides very high coding efficiency, especially at medium and low bit rates., where the proposed system outperforms both H.264 and the earlier MP video coder developed by Neff and Zakhor [87].

Chapter 5 RATE DISTORTION SOURCE MODELING FRAMEWORK FOR MATCHING PURSUIT VIDEO CODING

Matching pursuit has been applied into video coding as an alternative to transform coding and it shows promising performance in Chapter 4. To further improve its efficiency and enable precise rate control in it, a source model that well describes its character needs to be developed.

5.1 Source modelling for video coding

Source modelling is widely applied in video coding. It has been stated in Chapter 1 that the ultimate purpose of rate control in video coding is to optimize the video quality under the bit rate constraint. To this end, many rate allocation schemes [21, 96, 97] have been developed to control the use of bit budget. In these schemes, operational rate distortion models are developed to describe the rate and distortion changing trend so that bit budget can be allocated accordingly to optimize the results.

Operational rate-distortion techniques have been used to improve the performance of DCT-based video codecs. Such methods seek to distribute the limited number of bits across multiple coding units in order to minimize the distortion [98]. For the DCT case, the coding units may be individual macroblocks, groups of macroblocks, or entire video frames. In [79, 99], this unit is defined to adaptive in size. A control parameter such as the quantizer step size is used to set the rate-distortion trade off for each coding unit. An optimization method is then employed to find the set of control parameters which minimize the. Various examples of this technique have been published for the DCT based systems [23, 98]. Assuming each frame represents a single coding unit, then the rate-distortion trade off for each frame is a function of control parameter. The problem is thus to choose the control parameters which minimize some distortion metric for the frames of interest.

In principle, such optimization methods could also be applied to the matching-pursuit video coders. Some work has done to develop and use such model for MP codec [100]. However, because of the special atom searching and encoding mechanism, the work developed in [100] is not directly applicable to the coder developed in BUMP. A specific model that considers the quantization and encoding methods needs to be developed.

5.2 The distortion modelling for BUMP coder

An effective source model illustrates the relationship between the bit rate and the distortion. In the MP case, where the system recursively code atoms, it is rational to have both values related to the atom number. That is, to develop two functions:

$$D = f_1(A) \tag{5.1}$$

$$R = f_2(A) \tag{5.2}$$

With the above two functions, it is then easy to relate the distortion with rate, :

$$D = f_1\left(f_2^{-1}(A)\right) \tag{5.3}$$

where $f_2^{-1}(\cdot)$ is the inverse function of $f_2(\cdot)$.

In section 4.2.1, the special MP algorithm is reviewed, and the relationship between the atom amplitude value and the distortion can be easily developed. In the following discussion, the distortion is measured with sum of square error (SSE).

The idea of matching pursuits is to decompose a signal into a linear expansion of waveforms, which belong to a redundant dictionary of functions $D = \{\varphi_\gamma\}$, with $\|\varphi_\gamma\| = 1$ for all γ. MP is a recursive procedure in which, at each time of iteration, a function $\varphi_{\gamma n} \in D$ that best approximates part of the signal is chosen. So, after *m* iterations MP decomposes the original signal f into a sum of dictionary elements, so that

$$f = \sum_{n=0}^{m-1} \alpha_n \varphi_{\gamma n} + R^m f\,, \tag{5.4}$$

where a_n is the matching pursuit coefficient (or inner product) given by

$$a_n = \left\langle \varphi_{\gamma n}, R^n f \right\rangle \tag{5.5}$$

and $R^m f$ is the ***m***th order residual vector after approximating the signal in the direction of $\varphi_{\gamma n-1}$.

When applied in a video system, the input function f to the MP coder is a DFD frame, which is the motion compensation error. If no further encoding is to be carried out, then $\|f\|^2$ is the distortion of that frame. Here $\|\cdot\|^2 = |\langle\cdot,\cdot\rangle|$ denotes the function energy. Substituting (5.5) into (5.4), equation (5.4) can be rewritten as

$$R^m f = f - \sum_{n=0}^{m-1} \langle \varphi_{\gamma n}, R^n f \rangle \varphi_{\gamma n} \tag{5.6}$$

After m times of such recursive iterations, the distortion is reduced to $R^m f$. Consider the situation after the first atom is encoded, we have

$$f = \langle \varphi_{\gamma 0}, R^0 f \rangle \varphi_{\gamma 0} + R^1 f \tag{5.7}$$

The operation of taking inner product $\langle\cdot,\cdot\rangle$ ensures that the residue $R^1 f$ is orthogonal to the basis function $\varphi_{\gamma 0}$. Since $\varphi_{\gamma 0}$ has a unit norm, and here $R^0 f = f$, we have the following relationship

$$\|f\|^2 = \|\langle \varphi_{\gamma 0}, f \rangle\|^2 + \|R^1 f\|^2 = \alpha_1^2 + \|R^1 f\|^2 \tag{5.8}$$

From the similar process, the relationship below can obtained,

$$D_n = \|R^n f\|^2 = \|R^{n-1} f\|^2 - \alpha_n^2 = \|f\|^2 - \sum_{n=0}^{m-1} \alpha_n^2 \tag{5.9}$$

The distortion after n atoms can be calculated using the original distortion and the sum of the amplitude of the found atoms. In the BUMP system, the MERGE coder rearranges the atoms before actually sending them. A redundant number of atoms are found before the MERGE encoding is performed. Therefore, the values of α_n^2 are already available. This enables the distortion to be calculated with (5.9).

The above analysis reveals that the relationship between the atom number and the distortion after encoding can be obtained from the available information. This leaves the main difficulty of source modelling in developing the rate model.

5.3 The rate modelling for BUMP video coder

Rate control is an important part of practical video coding. Rate distortion theories model the relationship between the coding bit rate and the signal distortion. Rate control and bit allocation schemes are then used to control the usage of bit budget and to optimize the system performance under coding constraints [101]. Many rate estimation models have been developed for conventional transform coding schemes and applied to their rate control [21, 28, 102, 103].

Rate control is also a vital part for MP based video coders and has been the subject of some previous research [100]. The work mainly focuses on two parts: the first is how to adapt the quantization value in the video framework along the coding procedure. The second aspect is the modelling of the relationship between the number of coded bits and atoms.

However, due to the different nature of coding methods exploited in BUMP project, the work of [100] cannot be directly applied to our coding scheme. This chapter studies the characteristics of how the bit rate changes with the number of coded atoms. This is defined as the Rate-Atom (R-A) analysis. A framework for modelling the bit rate of the MP video coder is proposed. This model is then used to design an adaptive rate estimation algorithm which is able to effectively track the R-A slope changes and ultimately provide accurate estimation.

5.3.1 Analysis of the bits number change of BUMP

5.3.1.1 Review of atom coding in BUMP

The BUMP codec encodes atoms in a special manner and therefore its rate distortion characteristics need particular study. The general encoding procedure of an MP scheme is to recursively search for the best approximation of the image using one basis function chosen from the codebook each time. At each iteration, the identity of the chosen basis function, its position in the image to be encoded and the inner product value result. Such a result is defined as one atom [88]. In BUMP encoder, the coding can be briefly described as a three-step per atom process, *Initialize Compute a full set of inner products between the image and all bases in a codebook.*

Repeat:

1. Find an atom. Full 1D or 2D search or reduced complexity strategies are possible.

2. Image Update. Subtract quantized atom from image.

3. Repair Inner Products. Re-compute required inner products only in atom footprint.

Until distortion or bit rate criterion met.

In [88], the found atoms are quantized, and then encoded in a raster scanning order according to their positions. The quantized atom value, the atom position and the basis indices are encoded respectively with adaptive Huffman code. Since BUMP applies totally different quantization and encoding methods to the atoms, the characteristics of its final rate-distortion curve are very different from those of previous MP coders. To explain the bit rate changes in BUMP coder, its entropy coding part, the MERGE coder, is reviewed in the following.

MERGE is a Recursive Group Embedded Coding algorithm. As introduced earlier, each atom chosen by BUMP is defined by a position $(XPos, YPos)$ in the 2D data space with the magnitude A and 2D codebook basis index (B_x, B_y). To efficiently encode the atoms, Precision Limited Quantization (PLQ) is used as a top-down quantizer to quantize the magnitude of the atom.

With PLQ, the magnitude of an atom's amplitude A is quantized into a triple parameter set $\langle S, F, R \rangle$, where S is the sign , F is the First Significant Bit (FSB) of A ($F = \log_2 |A|$) , and R is the remainder subject to the value of PL in the range 0 to $2^{PL-1} - 1(PL > 1)$. The example illustrated in Figure 5.1 b) has a PL of value 4 (including the FSB).

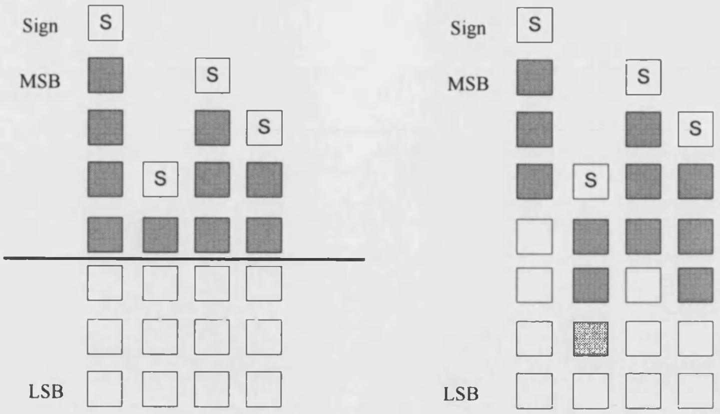


Figure 5.1 Uniform quantization and precision limited quantization

Lossless coding of the attributes is done by the MERGE algorithm, in which atoms are gathered into groups with all attributes in common, and the positions are signaled by run length coding. This scheme works well with PLQ, which keeps the number of groups reasonably small. To reduce the number of groups further, the sign *S* of each atom is sent as one bit of side information which is efficient because positive and negative signs have near-equal probability. If the Precision Limit is *PL*, the MERGE algorithm is:

```
For F (the FSB) from Maximum to Minimum
    For R (the amplitude Residual) from ( $2^{PL-1}-1$ ) to 0
        For each Basis Function K used
            Signal by Run Length Coding the position of each atom with attributes (F, R, K).
            Send the Sign S of the atom (1 bit)
        End of K (Basis Function) Group
    End of R (PLQ Residual) Group
End of F (FSB) Group
```

Maximum embedding is achieved by sending atoms in order of decreasing amplitude, with the codebook entry as the innermost loop. MERGE automatically compensates for variations in the frequency of occurrence of the attributes of the atoms and eliminates the need for entropy coding. The adaptive run length coding in MERGE adjusts to the statistics of the atom position.

5.3.1.2 Analysis of the Rate-Atom relationship in BUMP

In a BUMP system, encoding is performed in an atom-wise manner. As the coding process continues, more and more atoms are coded and the number of bits used increases. It is therefore rational to look into the relationship between the bits rate and the number of encoded atoms. In later part of this paper, we call it Rate-Atom (R-A) relationship.

Taking one frame from the test sequence Akiyo as an example, the coding process of that frame is displayed in details in Table 5.1, which lists some lines quoted from a log file recording how the bits are used along the encoding process of a BUMP based coder.

According to the MERGE scheme, there are different kinds of information to be encoded for each frame. From the table it can be seen that some bits are spent for coding the file header and group information, such as group header, and some others for coding the random empty groups. Also, the number of bits used for each position depends on the atoms' spatial distribution, and is generally random.

Looking into the procedure of encoding a picture, the following are the main factors affecting the ultimate bit rate:

i). the value of the amplitude of the atoms. Larger values take more bits to encoder;

ii). the distribution of data values. MERGE code data by dividing it into groups, and a sparse data distribution means there are more groups involved, so that more bits are needed to deliver the group changes;

iii). the number of atoms to be encoded. Higher target quality often requires more atoms to be coded to transmit enough information to enable an image to be constructed with more fidelity.

A plot of the R-A relationship during coding this frame is shown in Figure 5.2. For each step in Table 5.1, there is one corresponding R-A point. Because various kinds of information need to be encoded, the bit number increases a few times for delivering one single atom. As indicated in Figure 5.2, such a phenomenon causes a high bit usage for atoms in the earlier stage, but has less impact to the later ones.

Atom	Bits Cost	Total Bits Number	Information transmitted
0	*29*	*29*	*header bits*
0	*8*	*37*	*FSBgroup_size*
0	*6*	*43*	*empty groups*
1	*17*	*60*	*atom sent*
1	*3*	*63*	*side information*
1	*1*	*64*	*sign coded*
2	*22*	*86*	*atom sent*
2	*3*	*89*	*side information*
2	*1*	*90*	*sign coded*
2	*7*	*97*	*End of Group*
2	*9*	*106*	*empty groups*
3	*20*	*126*	*atom sent*
3	*3*	*129*	*side information*
3	*1*	*130*	*sign coded*
3	*3*	*133*	*End of Group*
3	*4*	*137*	*empty groups*

Table 5.1 Example lines quoted from encoding log file

The exact number of steps for coding each atom depends on the data distribution, which might vary from frame to frame and is hard to track. Therefore, when discussing the R-A relationship in the following sections, only the total bit cost after finishing coding each single atom is considered.

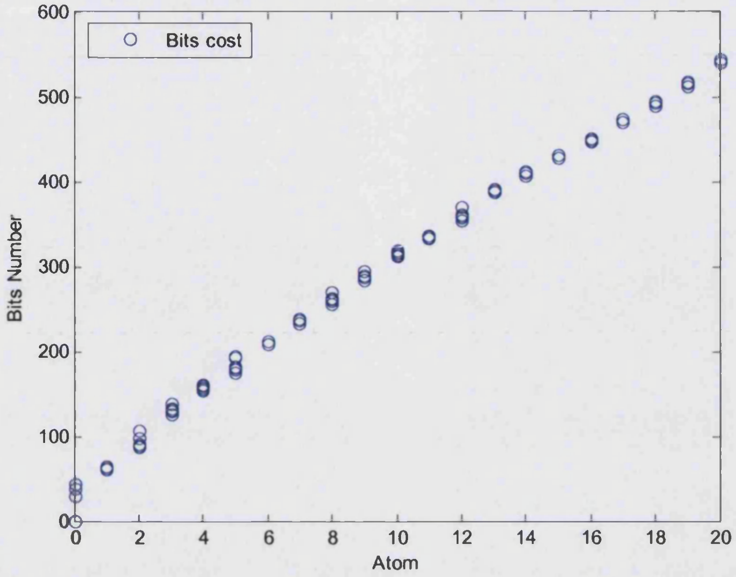


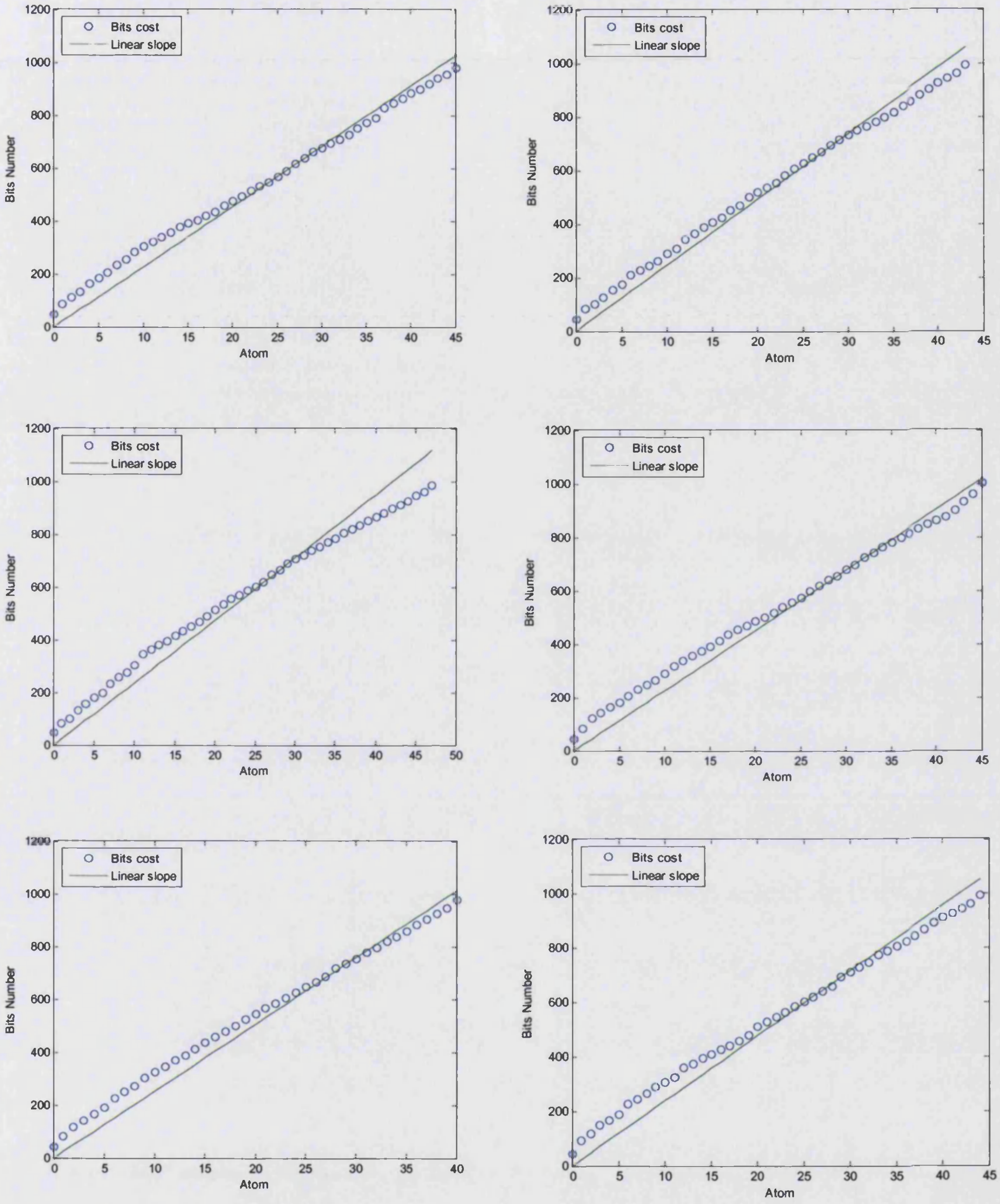
Figure 5.2 Bit increment along coding each atom

In the next experiment, to investigate the two test sequences, Akiyo and Foreman are encoded with the BUMP based video coder. Three frames of each are chosen and the Atom-Bit relationship is given in Figure 5.3. In this way, it can be seen that number of bits increases with the number of atoms and the plot shows a near linear relationship. To demonstrate such near-linear relationship, the standard linear functions are also shown in Figure 5.3. Based on the observation, the bits number change can be roughly modelled with a straightforward linear function,

$$R = k \times A, \tag{5.10}$$

where R is the bits number and A is the number of atoms that have been encoded.

This model from equation (5.10) has a very simple form, hence introduces very low complexity in calculation when being applied to rate estimation, but also has very obvious disadvantage: it gives very limited accuracy. The actual bit change highly depends on the frame content, and can show pretty large deviation from the linear line. As indicated in Figure 5.3, such inaccuracy is especially large over the earlier stage, where more bits are required for coding header information, and some later atoms, where the groups tend to have bigger sizes and fewer bits are need to deliver the atom information.



A) B)

Figure 5.3 The bit rate changes in 6 frames

A) Frames from Akiyo; B) Frames from Foreman

So, simply assuming a linear relationship is not enough to describe the trend of bits increments over the atoms. A more precise model is required to provide accurate bit rate estimation.

5.3.2 The proposed Rate-Atom modelling framework

To find a better rate estimation model, further analysis needs to be carried out. Figure 5.4 shows the coding process of another inter frame taken from the test sequence Hall. This time, we connect the point representing the last atom to the origin. It is seen that the R-A curve lies on top of the straight line, with convex on the upper side, as in Figure 5.4. This means the increase speed of bits number slows down as the coding procedure is going on. It is shown in wider experimental results that this phenomenon remains valid for all the sample frames. A curve with such character can be modelled with a power function,

$$f(x) = x^{\lambda}, \qquad \text{with } \lambda < 1 \tag{5.11}$$

Considering the fact that the bit rate increases with a near linear character, the model describing the relationship between the atom number and bits rate can be developed as

$$R = \theta \times A^{\lambda} + b, \tag{5.12}$$

This model in equation (5) comes in a very simple form. ***θ*** describes the general trend of the bit rate increment, ***λ*** defines how the curve deviate from the linear line, and ***b*** is introduced to take some extra bits for encoding side information into account.

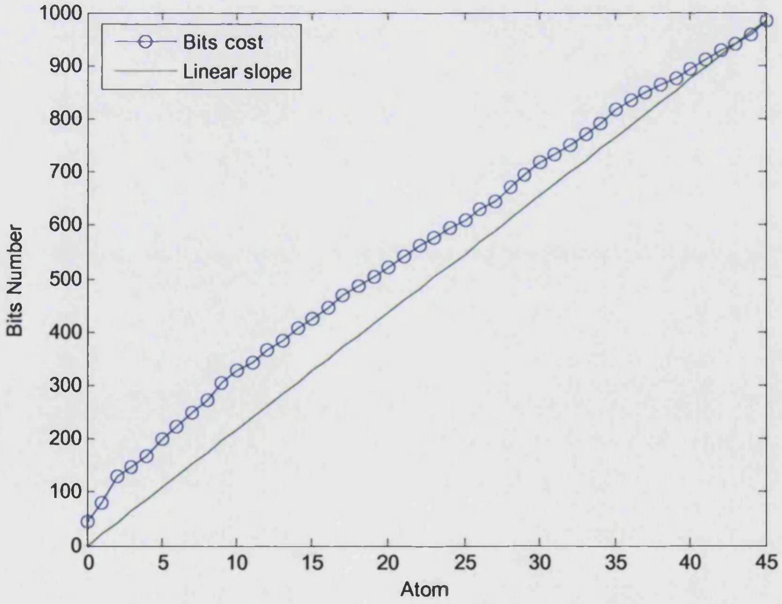


Figure 5.4 Deviation of the B-A curve from the standard linear plot

With a fixed λ value of 0.9, and ***b*** value of 0, the model is applied to some sample frames and the results are given in Figure 5.5. It can be seen the model fits the estimated line along the real R-A curve very well, and therefore, this model has much potential to be used for estimating the bit rate changes along the BUMP encoding process.

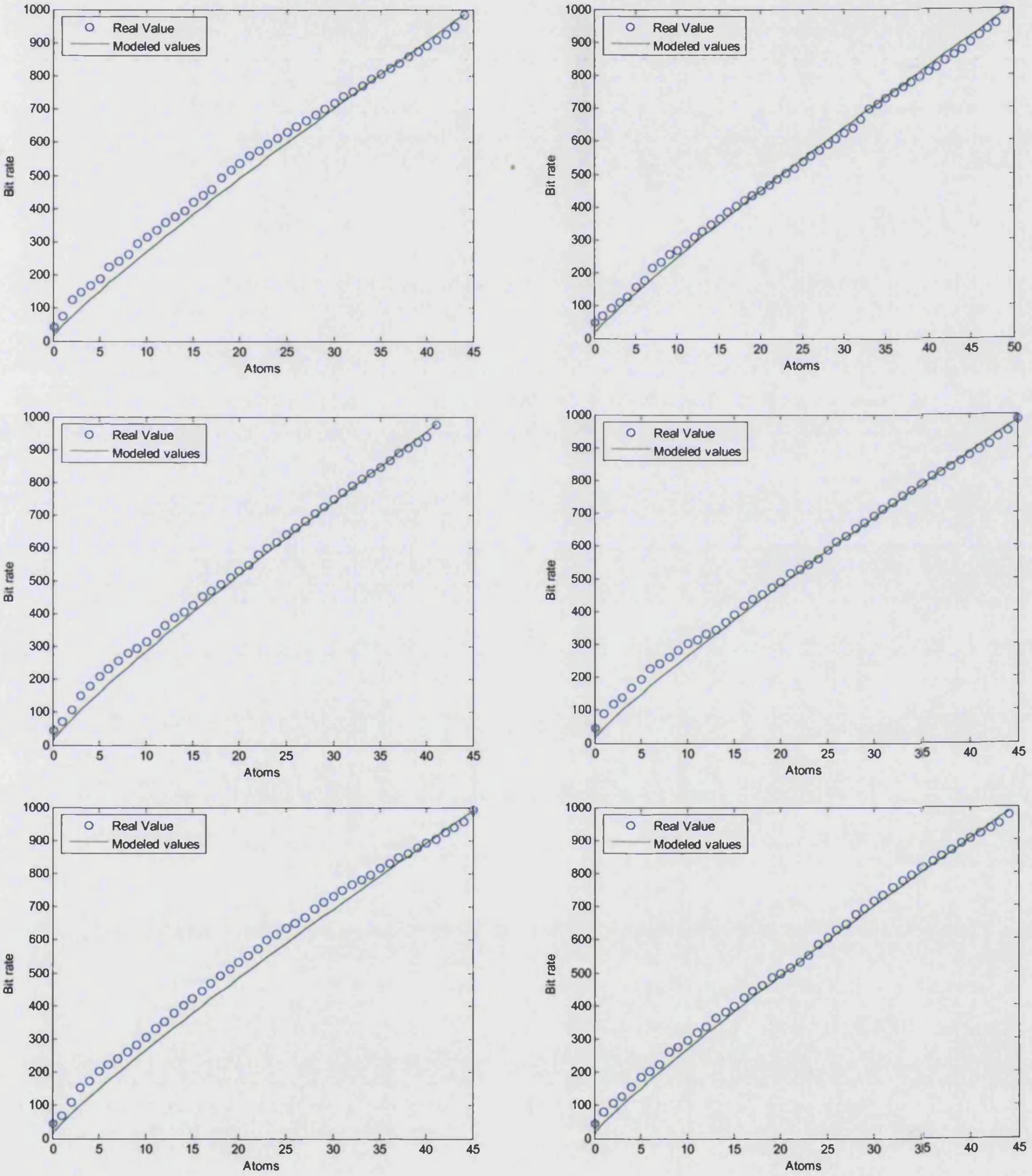


Figure 5.5 The modeled value achieved and the original data

(with λ=0.9, *b*=0)

The correlation coefficients of the modelled values and the actual bit rate are plotted in Figure 5.6, where eight frames are taken from each of the four different sample sequences. The chosen sequences include those containing complex motion, such as Stefan, and those involving very little movement, such as Akiyo. It can be seen that across the various styles of video content the correlation coefficient is very close 1, which proves that the model is accurate and robust to video contents variation, and therefore is able to effectively demonstrate the character of the R-A relationship.

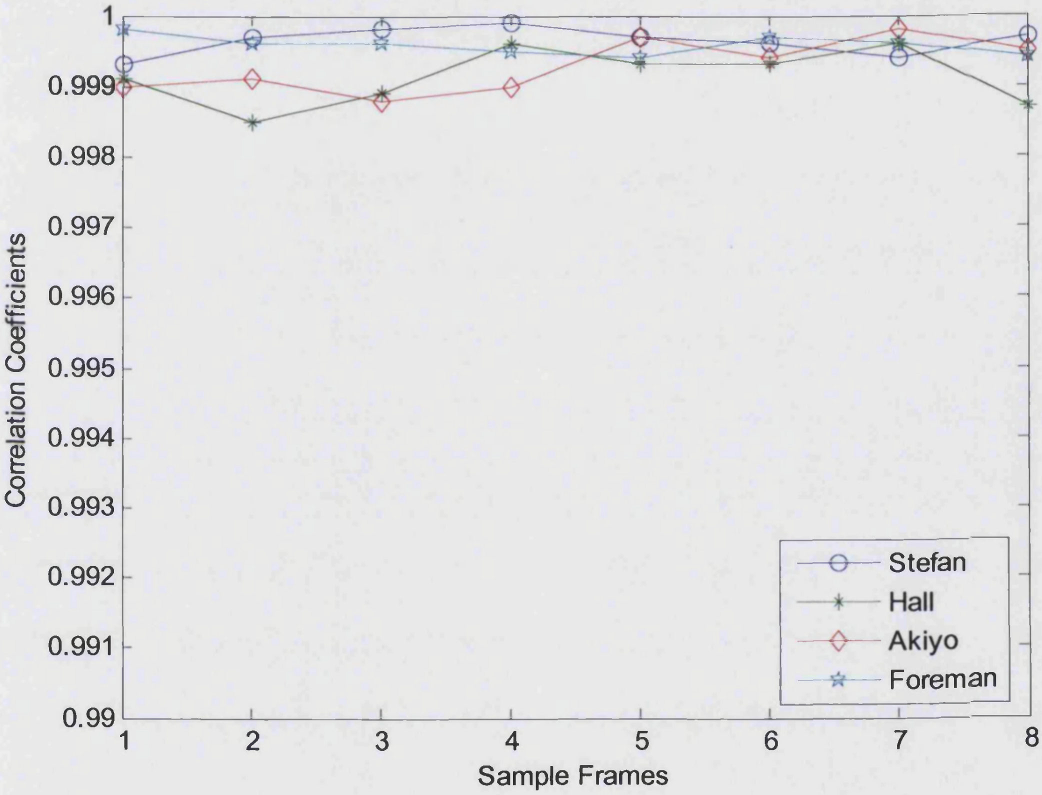


Figure 5.6 The correlation coefficients between the modelled values and the actual bits number

5.4 The adaptive bits estimation algorithm

The proposed framework accurately models the relationship between the coded atom numbers and the bit rate cost, and therefore is a powerful tool for operational bit rate estimation. In this section, we study how the framework parameters can be set to achieve accurate rate estimation.

5.4.1 Updating the proposed framework for bit rate modelling

Once the framework function is defined, the accuracy of the estimation mainly relies on the choice of the parameter values. In practice, the operational R-D modelling constructs the R-D curves by mathematical processing the observed data [101]. The parameters of the framework are calculated based on some encoding results of the visual data for compression. As the coding process progresses, the model's parameters need to be updated.

The proposed framework comes in a very simple form, and there are only three parameters to be calculated. The λ defines the shape of the R-D curve. It has been mentioned that the value for λ must be smaller than 1 to coincide with the real R-A curve, which shows a reduced speed of increase of bit rate as the atom number increases. θ describes the general near linear trend of the bit rate increment, and can be seen as a linear regulation factor to direct the model. The two parameters interact with each other and therefore their values need to be determined together. In Figure 5.7 we plot the modelled curves obtained from the framework with the same θ value and various λ values. The curve with λ value of 0.9 is a decent model for the rate of coding the sample frame. When λ varies, the resultant curves deviate significantly from the original one. θ is the regulator that makes sure the slope of the curve matches that of the actual data. To model different slopes, θ must be adjusted according to the λ.

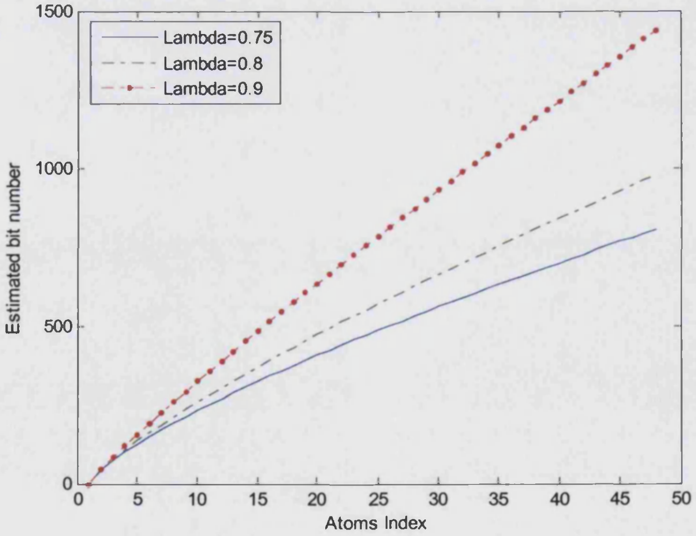


Figure 5.7 The modelled curves from the framework with same θ value and various λ values

To summarize, the model can be updated by calculating ***θ*** after the value of ***λ*** has been chosen. In this manner, the three curves in Figure 5.7 are all directed to follow the trend of the actual data, as shown in Figure 5.8.

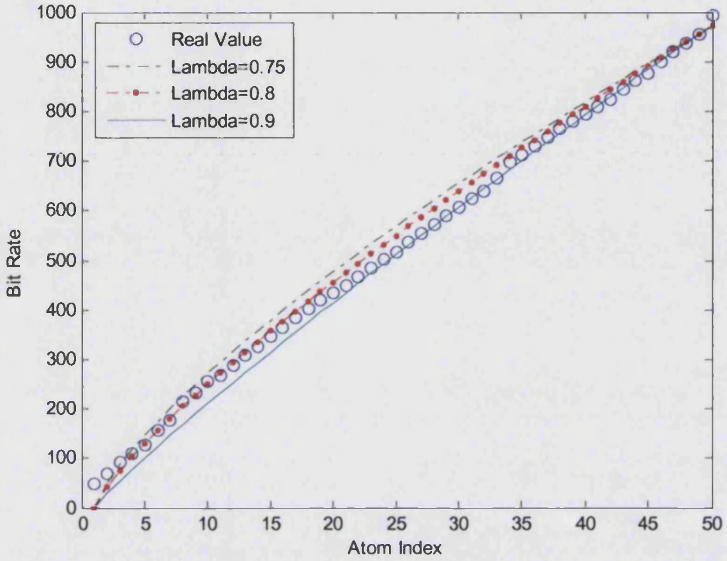


Figure 5.8 Rate estimations with different model parameter values

5.4.2. A rate estimation scheme based on the proposed R-A model

A closer study of Figure 5.8 indicates that the three modelled curves each simulate the actual data well for certain specific ranges while showing increased errors over others. This indicates that the curve shape varies over the encoding procedure, and any chosen curve shape is only valid within certain local ranges. Therefore, applying an adaptive parameter updating scheme which considers only the recently coded atoms might be able to track the shape changes closely.

Considering all the above characters of the visual signals and the framework, we propose the following bit rate estimation algorithm for BUMP based encoders,

Step 1). Initialize the model. Set a local "window" size of atom number. Carry out the BUMP frame encoding and keep the resultant bit rate until one "window" of atoms has been encoded. Keep the resultant bit number of those atoms for Step 2. In our experiment, the "window" size is set as 4.

Step 2). Update the model. Use the obtained bit number of the recently coded "window" size of atoms to compare with the candidate ***λ*** values. Choose the one that makes the modeled data the best estimation. The range for λ is set to [0.7, 0.95] in our experiments.

Step 3). Calculate the θ value for the framework with the estimated λ value, and collected bit number.

Step 4). Estimate the bit rate after coding the next atom with the two parameters above. If the atoms to be estimated is the last one in that frame, the parameter *b* must be introduced into the model, with an empirical value of b_0=15. The estimated bit rate R is therefore,

$$R = \theta \times A^{\lambda} + b_0,, \qquad \begin{pmatrix} b_0 = 15 & last\_atom \\ b_0 = 0 & else \end{pmatrix} \tag{5.13}$$

Step 5). Encode the next atom, record the resultant bit rate, and go back to Step 2).

The range of λ in Step 2 is defined according to a large number of experiment results. The R-A slope always shows a near linear relationship, and therefore the value of λ is always close to 1. When the encoding process meets the criterion, either the target rate or quality, and needs to be truncated, on top of those already used for coding the atoms, some more bits are required to encode the End of Group, and End of File signs, which makes it necessary to introduce the parameter of ***b***. The exact required number of bits varies, and an average value of 15 is used.

5.5 Experiments

The bit rate estimation scheme presented in section 5.4 is implemented and applied in the hybrid video coder with BUMP and H.264, which has been described in section III. To assess the accuracy of the proposed scheme, four video test sequences, namely Akiyo, Hall, Foreman and Stefan, are encoded with frame types being set as IPPPP. The first I frame is encoded with H.264 and all successive P frame are encoded with the BUMP inter frame coder. The rate estimation scheme is performed over the inter frames. For each frame, the estimation is recorded at each stage, and eventually plotted to be compared with the actual bits cost.

Since the "windowed" data collection focuses the parameter estimation on the recent coded atoms, it closely follows the actual changes happened to the R-A slope. Plotted in Figure 5.9 are the bit estimation result and the actual bit number for encoding the 5th frame from the Foreman sequence. It can be clearly seen that the model efficiently detects the changes in the R-A curve's shape, and gives a precise prediction. Figure 5.10 shows the relative error (RE) of the estimation, which is defined as

$$RE = \frac{Estimation\_Error}{Actual\_Bits} \times 100\% \qquad (5.14)$$

Since the model does not start estimating until 4 atoms have been encoded, we only plot the relative error from the 5th atom onwards. The estimation error quickly converges to near zero, and from the 13th atom onwards, remains within the range of ±0.02.

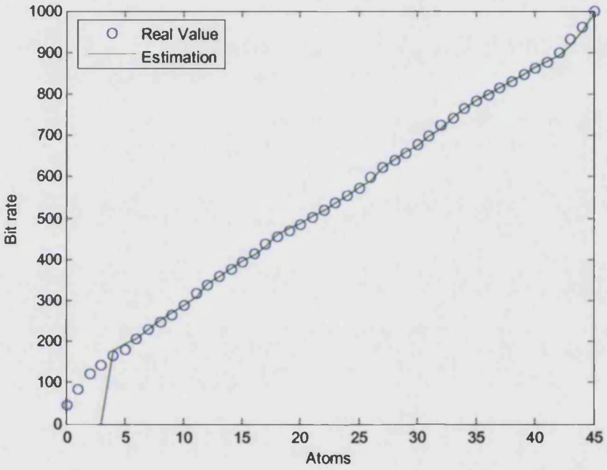


Figure 5.9 Comparison of bit estimation and the actual bit number for frame 5 in Foreman

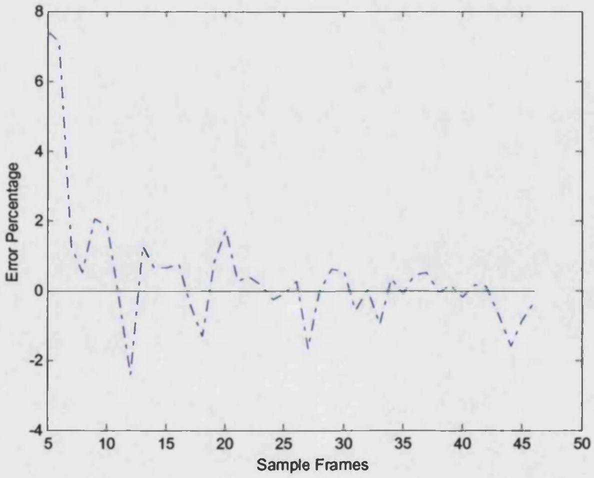


Figure 5.10 The relative estimation error for frame 5 of Foreman

Similar results are obtained from coding Stefan. The results for coding the 8th frame of Stefan are plotted in Figure 5.11 and Figure 5.12, where the algorithm also shows a very high accuracy. Although the relative error is nearly 6% at the 10th atom, it soon becomes smaller than 2%.

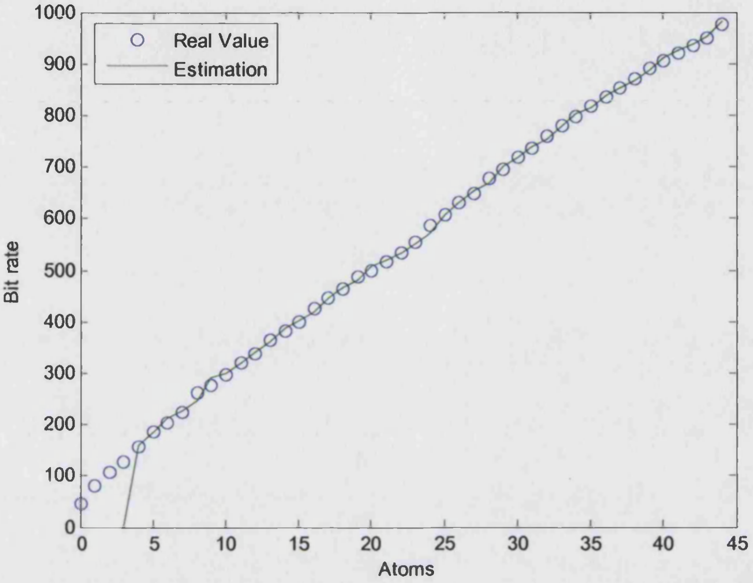


Figure 5.11 Comparison of bit estimation and the actual bit number for frame 8 in Stefan

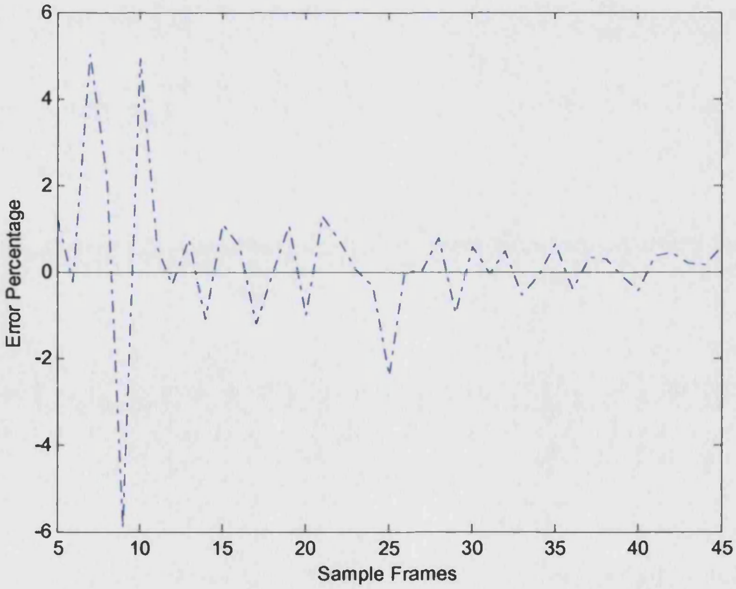


Figure 5.12 The relative estimation error for frame 8 of Stefan

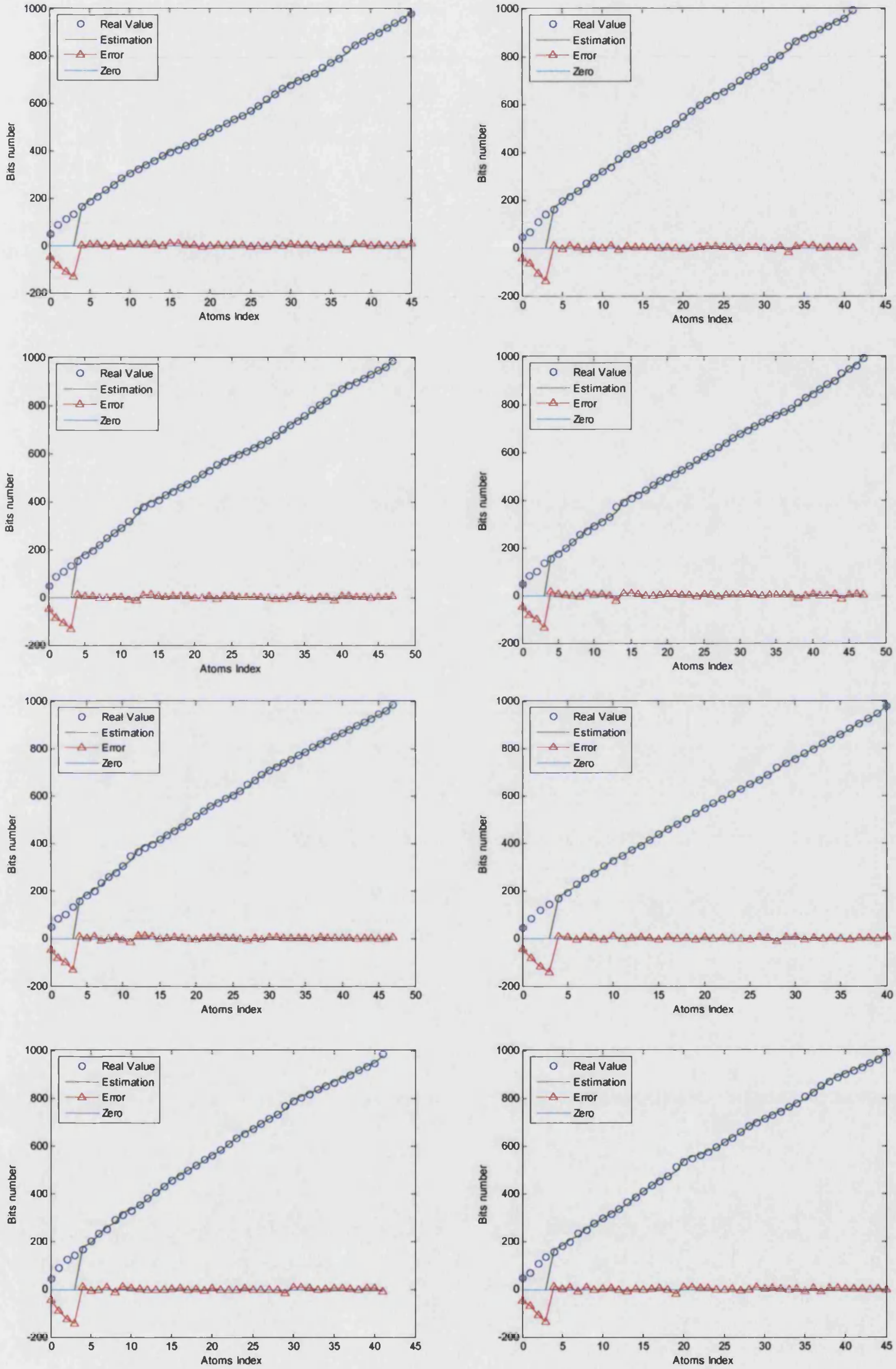


Figure 5.13 The estimation results and errors for sample frames from Akiyo

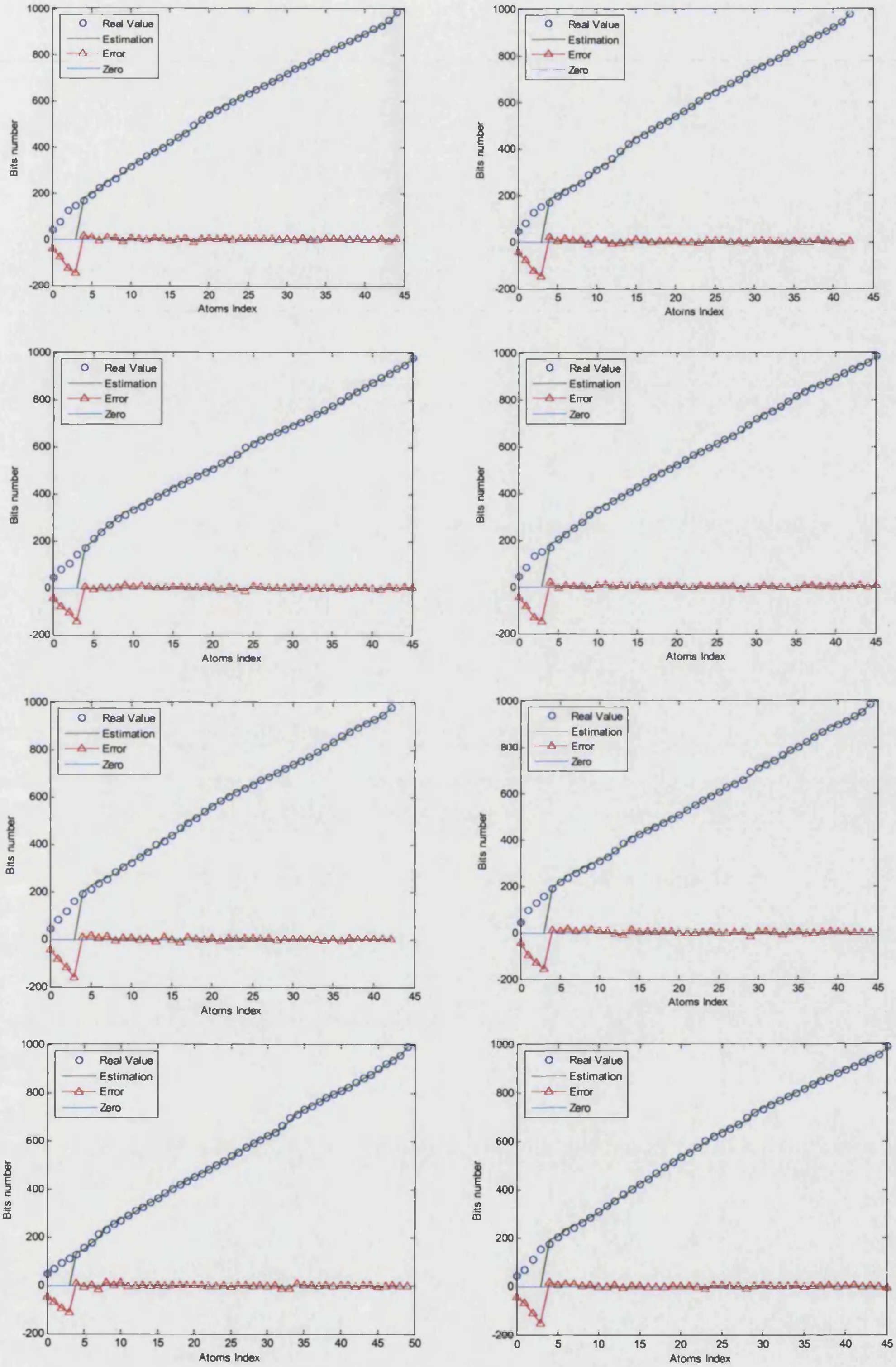


Figure 5.14 The estimation and errors for sample frames from Hall

The average values of more experimental results are given in Table 5.2. To ensure the accuracy in averaging, both Mean Error (ME) and Relative Error (RE) are calculated in absolute values, as defined below,

$$ME = \frac{\sum_{Estimations} |Estimation\_Error|}{Estimations} \tag{5.15}$$

$$RE' = \frac{\sum_{Estimations} \frac{|Estimation\_Error|}{Actual\_Bits} \times 100\%}{Estimations} \tag{5.16}$$

where *Estimations* refers to the total number of estimated values within this frame. From the data in the table, most RE values are smaller than 1.5%, which indicates very high prediction accuracy.

	Akiyo		Foreman		Hall		Stefan	
Frames	ME	RE(%)	ME	RE(%)	ME	RE(%)	ME	RE(%)
1	3.7361	*0.8435*	4.1021	*1.0343*	3.5220	*1.0070*	4.0141	*0.8887*
2	4.4163	*1.0863*	4.1422	*0.8534*	4.0210	*0.8851*	3.1203	*0.7893*
3	3.8639	*1.0023*	4.0148	*1.0046*	4.3148	*1.1369*	4.1084	*1.0870*
4	4.3485	*1.0437*	3.2557	*0.8449*	3.2250	*0.9555*	3.8585	*1.0901*
5	4.5462	*1.0399*	3.8750	*0.9899*	4.2818	*1.2454*	3.2926	*0.8596*
6	4.4115	*1.1256*	3.8693	*1.0590*	3.5384	*0.9084*	3.8965	*0.9831*
7	3.6539	*0.8851*	4.8866	*1.3028*	4.4915	*1.1269*	4.2381	*1.0668*
8	4.6096	*1.1297*	3.5869	*0.8668*	3.4139	*0.9264*	3.9780	*0.9626*

Table 5.2 The average errors from estimating sample frames of the 4 test sequences

5.6 Conclusion

In this chapter, a rate-distortion modelling framework for Bath University Matching Pursuit video coder has proposed.

The mechanism of MP allows a straightforward distortion measure at encoding stage. Considering the special quantization scheme of PLQ employed in BUMP system, an efficient distortion estimation scheme is presented. A detailed theoretical justification for the unique property of the R-A curves in BUMP encoder is provided. Based on this modelling framework, an adaptive rate estimation algorithm has been proposed. Experiments show that the proposed algorithm effectively tracks the variations of the R-A curve shape during the encoding procedure and finally provides estimation with a surprisingly high precision. Together with the powerful rate estimation framework presented above, a Rate-Distortion optimization scheme for BUMP is to be developed in the next stage of this project.

There are three major contributions of this work. First, the encoding mechanism of MP is analyzed and with consideration of the PLQ quantization scheme employed in BUMP, an effective distortion estimator is proposed. Second, a novel Rate-Atom analysis methodology for a matching pursuit coding scheme which has a novel approach with very low complexity and extremely high accuracy, is presented. Finally, an efficient rate estimation algorithm is proposed based on the framework. Experiments show that this algorithm is robust to video contents variations and able to effectively track the localized R-A curve shape changes and provides very high ultimate accuracy.

Chapter 6 RATE DISTORTION OPTIMIZATION FOR MATCHING PURSUIT VIDEO CODER WITH RATE DISTORTION SLOPE CONTROL

6.1 Introduction

The mechanism of BUMP coder enables it to produce a highly scalable bit stream. And the rate and distortion model for BUMP coder, as presented in Chapter 5, guarantees a very high precision in estimation. To this end, any target bit rate or quality at frame level can be easily achieved. This ultimately allows very effective rate and quality control.

However, to optimize the performance of the video coder, extra manipulation must be considered. The technique presented in Chapter 4 is ad hoc, and the experiments are performed on an extreme GOP structure, which is not likely to be used in practice. In this chapter, the more general case is to be studied. Rate distortion optimization requires knowledge of both rate and distortion character of the input, both of which can be calculated with the model presented in Chapter 5. In this Chapter, the model is applied to optimize the BUMP video coder's performance.

6.2 Optimization algorithm: Rate Distortion slope unifying

In Chapter 3, two optimization schemes are presented. The R-D slope unifying scheme is inspired by the idea of Lagrange multiplier, and it is a proper option to be applied to optimize BUMP coder.

In Lagrangian multiplier method, a cost function is defined to measure the R-D performance of a selected parameter setting:

$$J = D + \lambda R, \quad (R = \sum R(n) \; D = \sum D(n)) \qquad (6.1)$$

where *D(n)* is the resultant distortion when coding the nth unit, and R(n) is the bits consumed for coding the *n*th unit, and λ is called Lagrangian multiplier. The process of optimization is to find a proper setting so that, together with R the resultant *D* minimizes the cost function *J*. The resultant optimal *R* changes with the Lagrangian multiplier *λ* and so λ can be changed to make the coding meet the target bit rate.

For each coding unit I, the point on the R-D characteristic that minimizes *J* is that point at which the line of the absolute slope λ is the tangent to the convex hull of the R-D characteristic [64], i.e.,

$$\frac{\partial J}{\partial R}=\frac{\partial D}{\partial R}+\lambda\frac{\partial R}{\partial R}=0, \qquad \lambda=-\frac{\partial D}{\partial R} \tag{6.2}$$

The Lagrange multiplier optimizes the coding result by achieving a constant λ in all the coding units of the sequence, and it is also called constant slope optimization. It has been shown in Chapter 3 that slope matching scheme is not appropriate to be applied to conventional coding schemes. The conventional video coding, the bit rate changes with the QP and two QP values need to be applied to calculate one slope value, which induces more calculation. Another difficulty is due to the non monotonic slope change at MB level.

The slope actually defines the edge return of the encoding. For BUMP coder, the edge return is the brought by the last coded atom. This can be calculated with the distortion the atom reduces, and the number of bits for coding it. This slope only involves one atom, and is easy to calculate with the rate and distortion models presented in Chapter 5. In the following part, the slope unifying scheme is applied into the BUMP coder for optimization.

6.3 Slope calculation and prediction

The R-D slope is defined as $-\frac{\partial D}{\partial R}$. Since the proposed method only tries to achieve a unified slope value, there is no need to keep the sign. So the slope value of interest is

$$\lambda=\frac{\Delta D}{\Delta R} \tag{6.3}$$

where ΔD_n is the distortion reduction by encoding the last atom, and ΔR_n is the bits used to code it.

6.3.1 Calculation of the distortion change

The distortion deduction is

$$\Delta D=D_{n-1}-D_n \tag{6.4}$$

By replacing the D_n with (5.9), the equation (6.4) can be rewritten as,

$$\Delta D_n = D_{n-1} - D_n = \left\|R^{n-1}f\right\|^2 - \left\|R^n f\right\|^2 = \left\|R^{n-1}f\right\|^2 - \left(\left\|R^{n-1}f\right\|^2 - \alpha_n^2\right) = \alpha_n^2 \qquad (6.5)$$

However, the accuracy of the above calculation may be deducted by two factors. The first is the quantization noise. Since the atom amplitude must be quantized before it is encoded, the value would then deviate from the origin, and therefore does not follow the rule of orthogonality in atom searching. This would even affect the accuracy of the atoms in the following searching procedure. Fortunately, the effect of quantization has already been compensated in the atom searching stage. Quantization is performed immediately after an atom is found. And the residual after extracting this atom is computed based on the quantized atom value, which makes sure that the atom searching in the following stage is not affected in sense of accuracy.

The second potential problem is the sequence of atoms to be encoded. In general, the MP algorithm tries to find the part of information that would help reduce the distortion most, and the atom amplitude would decrease as the encoding carries on. However, this is not a monotonic change. There are cases that large valued atoms appear later. (Figure 6.1 to Figure 6.4). While when the atoms are coded, BUMP tries to produce the optimal embedded bit stream, and the special looping operation in MERGE makes sure the atoms are encoded following a descending order of the amplitude. This would then shuffle the original order in which the atoms are found. The construction of the frame therefore might suffer from some deviation from the ideal values.

A deeper study on this problem would ease such concerns. There are only very few cases of such wrongly-ordered atoms, so the affect would not be large. And in practice, from the view of the decoder side, the reconstruction of the frame is just a blind operation of summation. The more atoms available, the higher the accuracy can be achieved. A simple adding processing would not be affected by the change of the element order. Although in the case of MP coding, the result at some medium stage might vary, the distortion still reduces following the atom amplitude, with very little errors.

About distortion calculation, BUMP also shows specific advantage. The PLQ scheme in BUMP quantizes the amplitude with a fixed bits number rather than a fixed division step size. Experimental results suggest an effective empirical value of PL=2, which means each amplitude value only needs to be refined to one bit after the most significant bit.

As the BUMP loop over FSB values, there are only two possible quantized values for that loop.

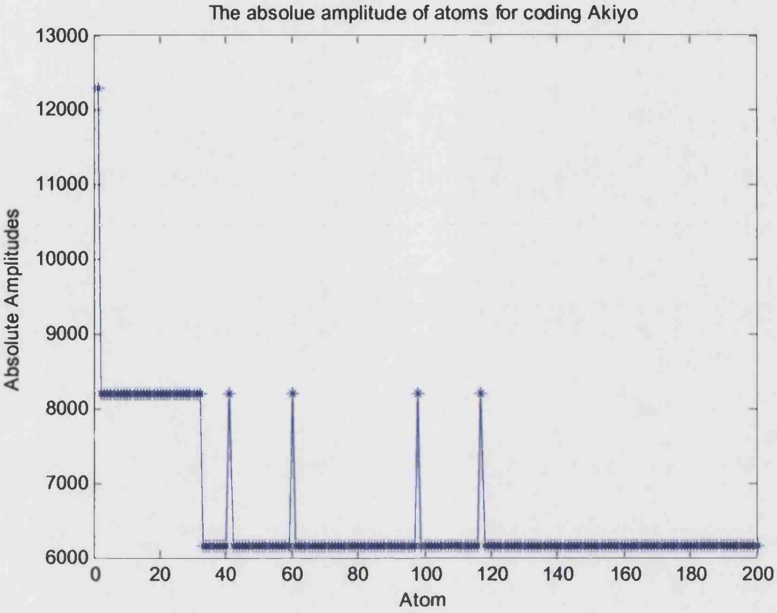


Figure 6.1 The amplitude of the first 200 atoms for coding a frame of Akiyo

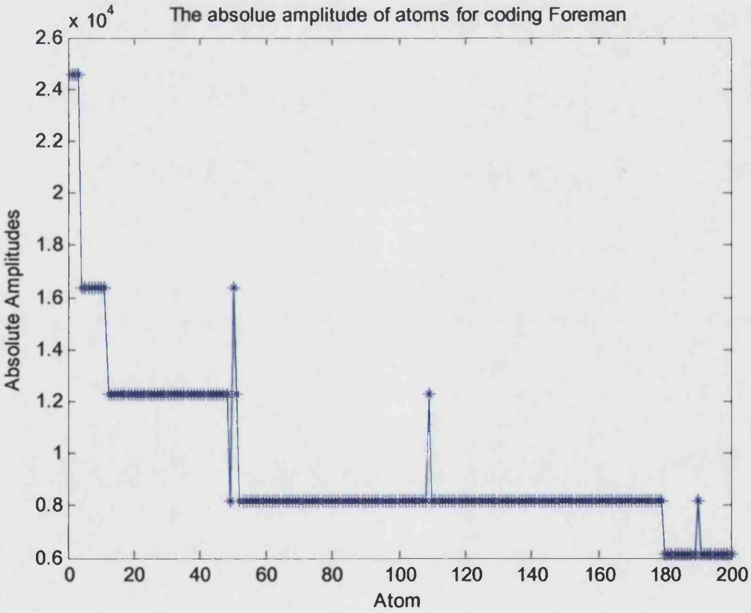


Figure 6.2 The amplitude of the first 200 atoms for coding a frame of Foreman

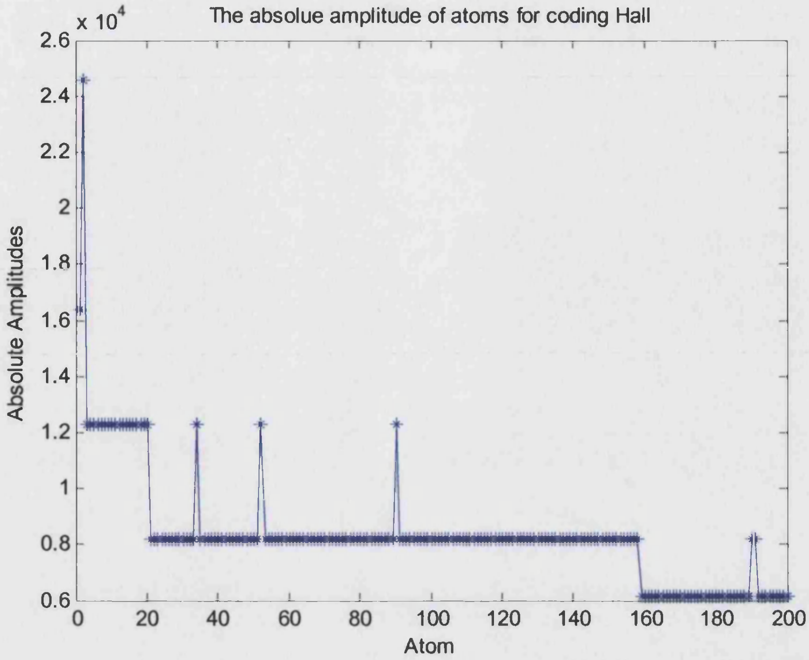


Figure 6.3 The amplitude of the first 200 atoms for coding a frame of Hall

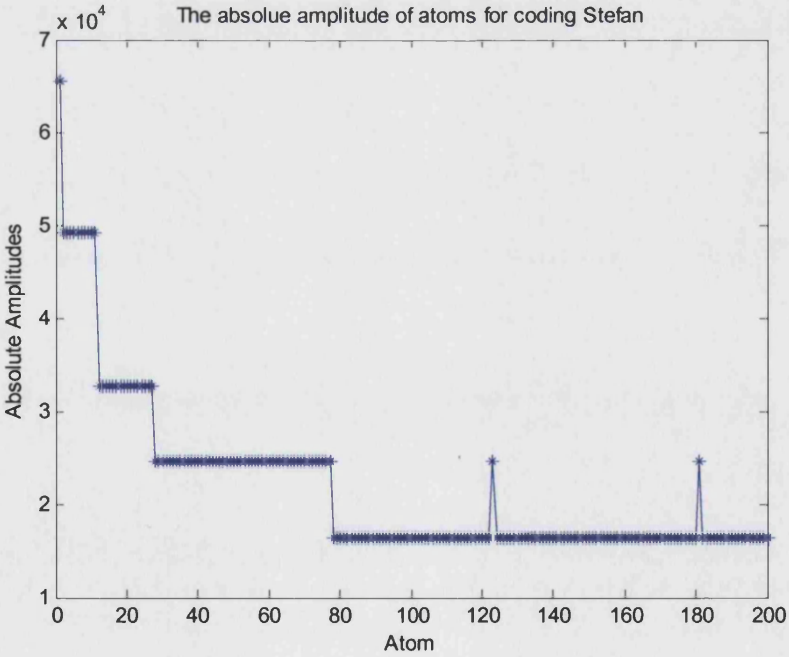


Figure 6.4 The amplitude of the first 200 atoms for coding a frame of Stefan

Figure 6.1 to Figure 6.4 show the values of the first 200 atoms for encoding four sequences. It can be seen the amplitude often takes one same value for a succession of atoms due to PLQ quantization scheme. As has been analyzed earlier, there are a few large valued atoms found after smaller ones. These are some rare cases and would not have much impact to the accuracy of distortion computation. All the amplitude values are taken from a very limited number of candidates and in practice, this makes the implementation easier.

6.3.2 Calculation of the bit rate change and the R-D slope prediction

To calculate the slope, the bits number change is also needed. The number bits used for coding the last atom is the bit change. So,

$$\Delta R = R_n \tag{6.6}$$

R_n is the bits used for encoding the *n*th atom. Since the slope calculation is for finding the truncation point of the output bit stream. When counting the bits number R_n here, the encoding is not finished. It has been introduced in Chapter 5 that at the end of encoding a frame, some more bits need to be used to code the End-of-File(EOF) sign. An empirical value of 15 is used in the model of (5.13). When computing the slope value, it is assumed that the sit stream is truncated at that point, and the EOF sign must be considered. So here (6.6) is changed to,

$$\Delta R = R_n + 15 \tag{6.7}$$

And (6.7) is used for finally calculating the slope.

The most useful application of the rate model is to predict the bit usage of future atom, which in this case can be used to predict the future R-D slope. For the same reason, when using the model, the EOF must be counted,

$$R = \theta \times A^{\lambda} + 15, \tag{6.8}$$

The BUMP finds redundant atoms before the actual encoding is performed, so the amplitude values of the atoms are available and ΔD can be worked out. Now the slope of the last atom and the future atom can be calculated and used for the purpose of calculation.

6.4 The rate and distortion optimization scheme with the source modelling

The BUMP coder produces a highly scalable bit stream and the optimization aims to find the best truncation point to cease the coding.

6.4.1 Algorithm overview

The algorithm applied here is similar to the slope unifying method proposed in Chapter 3. It can be briefly depicted as following.

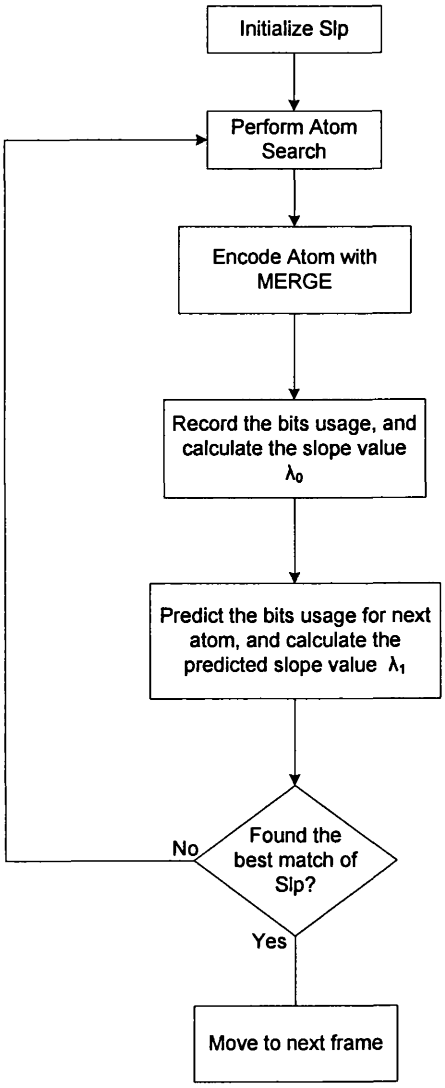

Initialize Slp
Perform Atom Search
Encode Atom with MERGE
Record the bits usage, and calculate the slope value λ_0
Predict the bits usage for next atom, and calculate the predicted slope value λ_1
Found the best match of Slp?
No
Yes
Move to next frame

Figure 6.5 The Slope matching optimization for BUMP video coding

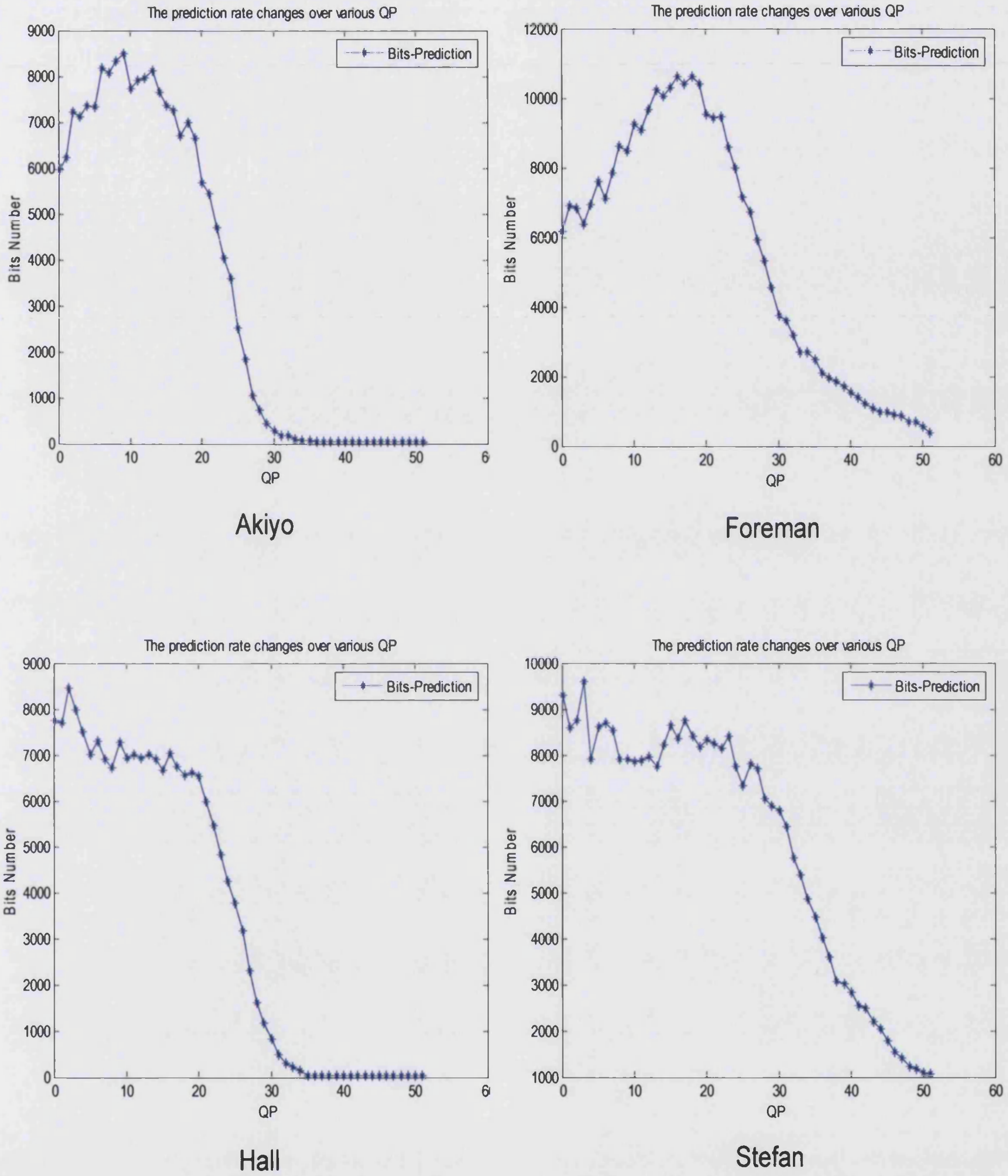
The rate model enables the slope value to be predicted. By studying how well the target slope is matched by the two calculated values, decision can be made on whether it is necessary to carry encoding to next atom. The sixth stage in the Figure 6.5 is a key part. The basic principle here is to compare the adjacent two slopes. If the predicted slope value λ_1 is a better match to the target *Slp* than the current slope λ_0, the encoding must continue. Otherwise, the bit stream can be truncated and the coding should be ceased.

Although the slope calculation here is actually performed on the frame level, the value might not always change monotonically, especially in the earlier stage. Therefore the criteria for making that decision must be carefully set.

6.4.2 The quantization parameter consideration

The bits used for coding an inter frame comprises two parts, the bits used for delivering the modes and motion vectors, and those for encoding the DFD. In the proposed rate control scheme, the focus is put on the latter and only this part can be controlled. However, since the original motion module is retained in the proposed hybrid video coder, and a QP must be chosen for that part. The choice of this value must also be considered when performing this rate control.

In original H.264, the rate is solely controlled by the QP value. The bit allocation over the two parts, the modes (including the prediction) and the residue, is judged and solved by the RDO feature, which takes the QP value into account. Therefore, the number of bits used for coding the prediction and modes is also controlled by QP. In Figure 6.6, the relationship between the QP and the prediction bits is depicted. These results are obtained from the original H.264 coder. It is shown that QP has a rather big impact to the bits number. The bits number generally decreases as the QP goes up. But relationship is not always monotonic. Both Akiyo and Hall contains very little motion content, and therefore the distortion contained in the residue frames is small, so the bits reaches the minimum value when QP is around 35. While for Foreman and Stefan, in which a lot more motion content is involved, the bits for prediction keeps decreasing.



Akiyo

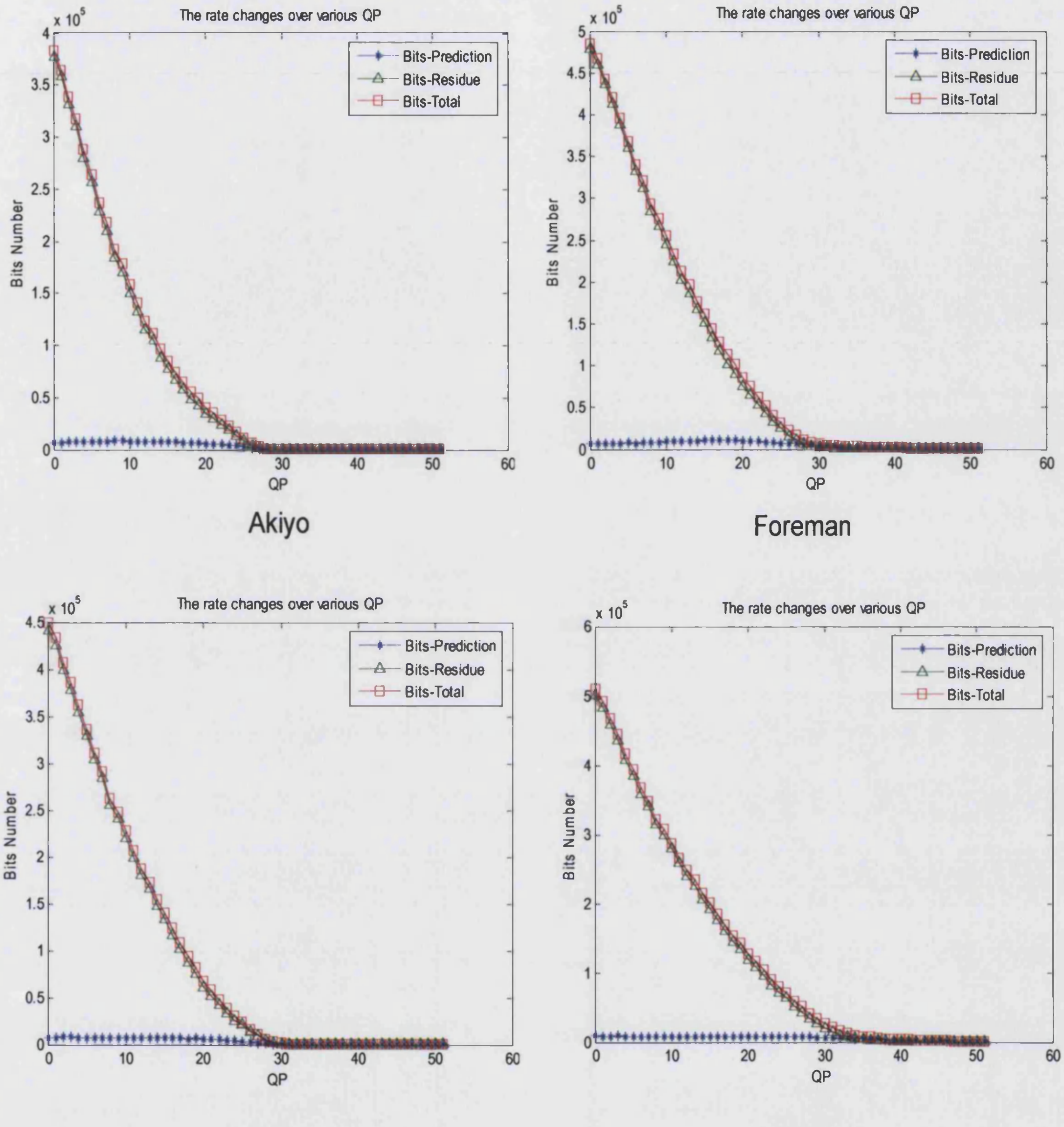
Foreman

Hall

Stefan

Figure 6.6 The bit number used for prediction for various QPs

From Figure 6.6, the impact of QP to the prediction bits seems to be quite large. But when it is plotted with the total bits in Figure 6.7, it is shown that this part is of only a very small proportion of the total bit usage. Since the bits number will be controlled by the proposed scheme, the QP does not have very big impact on the ultimate bit rate.



Akiyo

Foreman

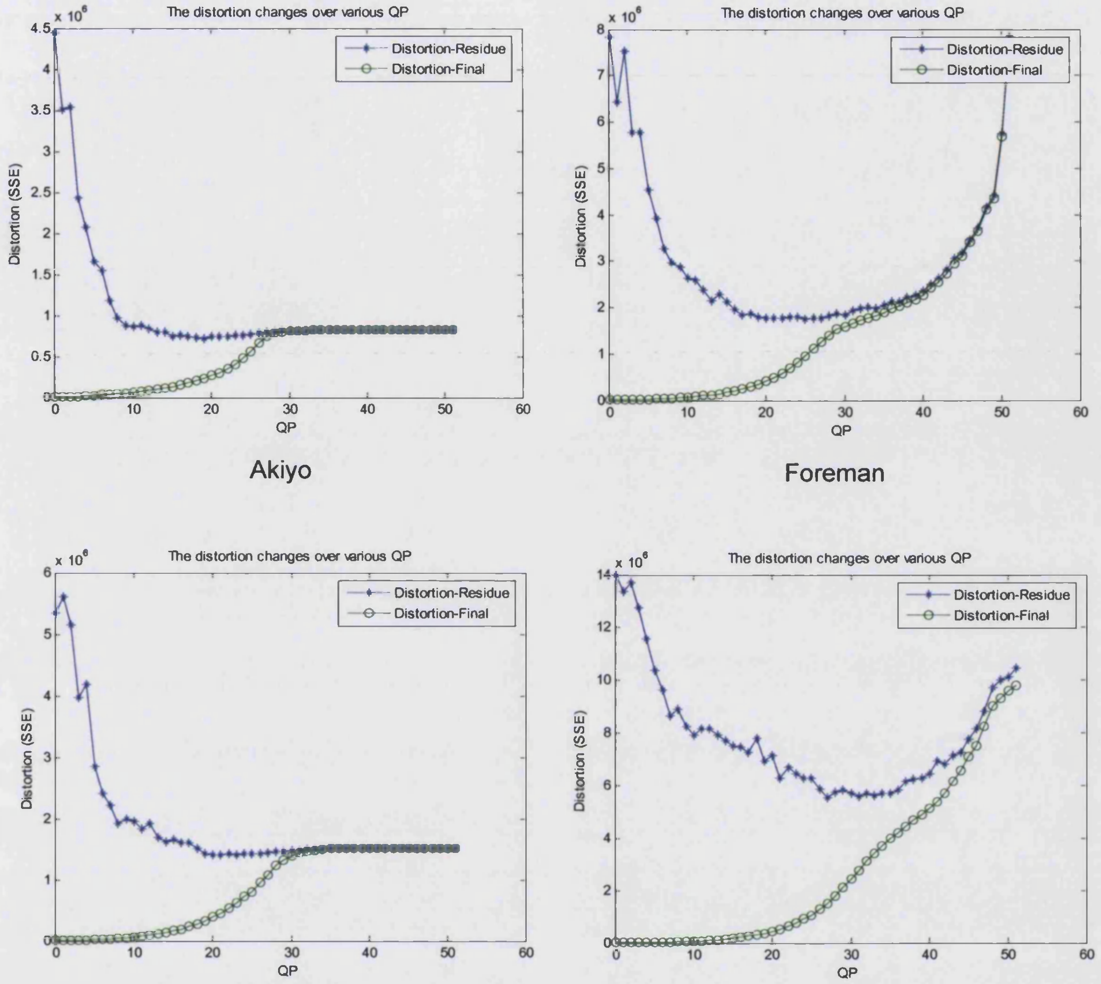
Hall

Stefan

Figure 6.7 The bits usage after different stages for various QPs

When QP goes up to about 35, the total bit rate is normally quite low, and the bits for prediction become the majority part of the total bit number. Only at such stage does the QP show a dominating effect on the final bit rate. It can also be expected that at such stage the QP would also have a major effect on the image quality.

Figure 6.8 depicts the distortion change when different QP is applied. With a QP of small value, the distortion value is deliberately left high so that the main distortion reduction is achieved by residue frame coding.



Akiyo

Foreman

Hall

Stefan

Figure 6.8 The distortion after prediction and the full coding for various QPs

The above results show that the setting of QP value would not affect the ultimate bit rate much in the BUMP system, especially when the target bit rate is high. For the residue distortion, the QP has a much higher impact. It should still be chosen following the traditional sense, a small QP for high bit rate.

6.4.3 The proposed bit allocation scheme

The method illustrated in Figure 6.5 could then be refined into with three extra features,

A. The motion estimation and compensation model must be tuned with proper QP value choice. To code a video sequence at a high bit rate, the target slope should be set small, and the QP value should also be small. There is a recommended empirical equation used in H.264 reference software to calculate the Lagrangian multiplier corresponding to the given QP value,

$$\lambda_{MODE} = 0.85 \times 2^{(QP-12)/3}, \tag{6.9}$$

$$\text{and, } \lambda_{MOTION} = \sqrt{\lambda_{MODE}} \tag{6.10}$$

Since the target slope value originates from the idea of Lagrangian multiplier, the above equations can be used. The difference between (6.9) and (6.10) is due to the measure of distortion. When deciding the coding mode, SSE is used, while for motion prediction, SAE is used. Since the proposed scheme also calculates the R-D slope with SSE, (6.9) is then used for deciding the QP.

The H.264 standard allows QP to be an integer value within the range of [0,51]. So the QP for tuning the motion module can be obtained as,

$$QP = \max\left(0, \min\left(51, \log_2\left(Slp/0.85\right) \times 3 + 12\right)\right) \tag{6.11}$$

where *Slp* is the target R-D slope value.

B. In the adaptive rate prediction method proposed by Chapter 5, the bits usage is predicted from the bits of the most recent four atoms. This means the earliest four atoms can not be predicted. This is solved with the following means.

1. The bits number for the first atom is not predicted. The starting stage needs to code a lot of side information so the exact bits number is very content dependent and hard to predict. In the method, the R-D slope of the first atom is only calculated after the atom is encoded.

2. To simplify the prediction for the next 3 atoms, empirical values are given to the model's parameters. For equation (5.13), λ=0.9 and ***b***=15 are used. $\boldsymbol{\theta}$ is updated from the bits of the previous atom.

C. The proposed optimization is to decide when to truncate the encoding. Ideally, if one atom delivers the target slope, the encoding should then be stopped. However, the calculated slope is a real number, and exact matching is almost impossible in practice, and a proper criterion is required for deciding when to cease coding.

At the stage of one finished atom, the scheme obtains the slope of this atom, Slp_0, and the predicted slope value Slp_1 for next atom. In general, the R-D slope value decreases along with the coding carries on. So if $Slp_0 < Target$, the coding should be ceased. In this case, carrying on coding might just cause the actually slope turns further from the target. Also, the slope change may not be monotonic sometimes. In this case, if such fluctuation happens with values very near to the target, the one that is closer to the target should be chosen. So, finally the condition is set as,

($Slp_0 < Target$ || $|Slp_0 - Target| < |Slp_1 - Target| \& \& |Slp_0 - Target| \leq |Target|$ || All atoms are coded).

The coding should be ceased when the above condition is true.

6.5 Experiments

The optimization scheme is implemented and adopted in the framework. Experiments are carried out to test its effect. In the experiments, four video test sequences, Akiyo, Foreman, Hall and Stefan are encoded. The GOP structure is set to be one that is close to those used in practice. One I frame is followed by 5 P frames.

6.5.1 The results of the slope based optimization

From (6.11), it is shown that the QP values correspond to target slope values of a very wide range. In the experiments, the target R-D slope values are taken from 1 up to 100,000. Figure 6.9 to Figure 6.12 the resultant average bit rates are plotted against different slope values. Because of the very big variation, the coordinate of target slope is presented with logarithm measure.

It is clearly shown that the small valued slope correspond to high bit rates, and vice versa. The exact shape of the plot depends on the video content, and there is dramatic change around the centre of the abscissa, corresponding to target slope value from under one hundred to a few hundred. The relationship between the target slope and the PSNR is plotted from Figure 6.13 to Figure 6.16, where the plots show near logarithm character.

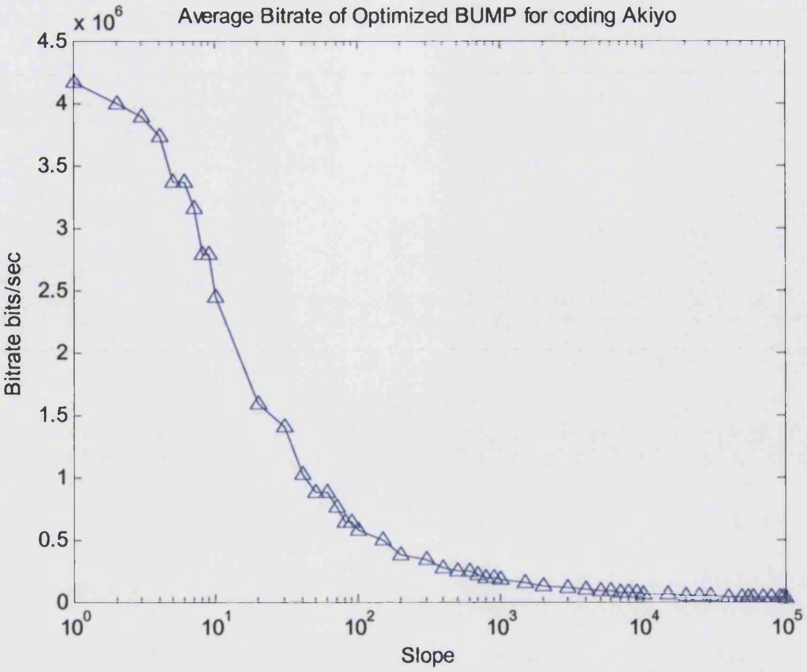


Figure 6.9 Resultant bit rates achieved for coding Akiyo

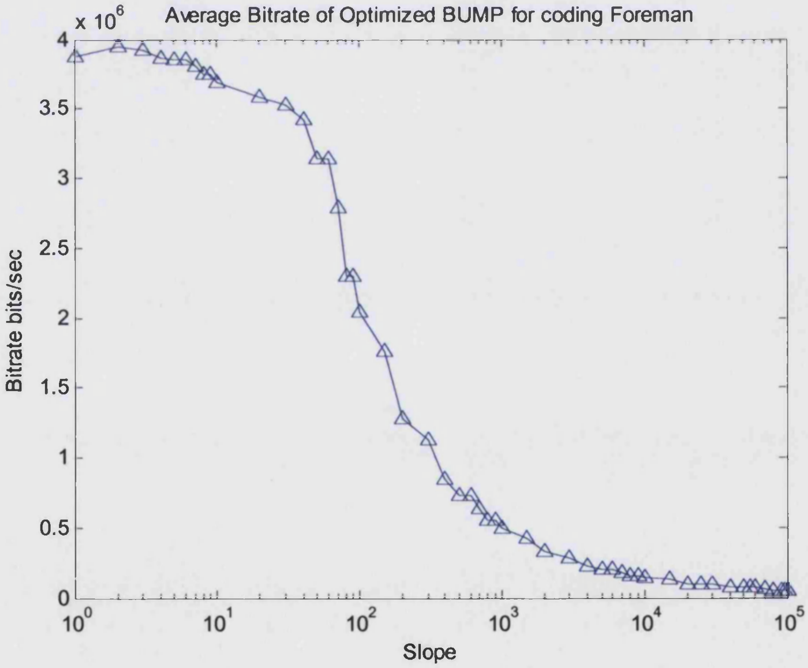


Figure 6.10 Resultant bit rates achieved for coding Foreman

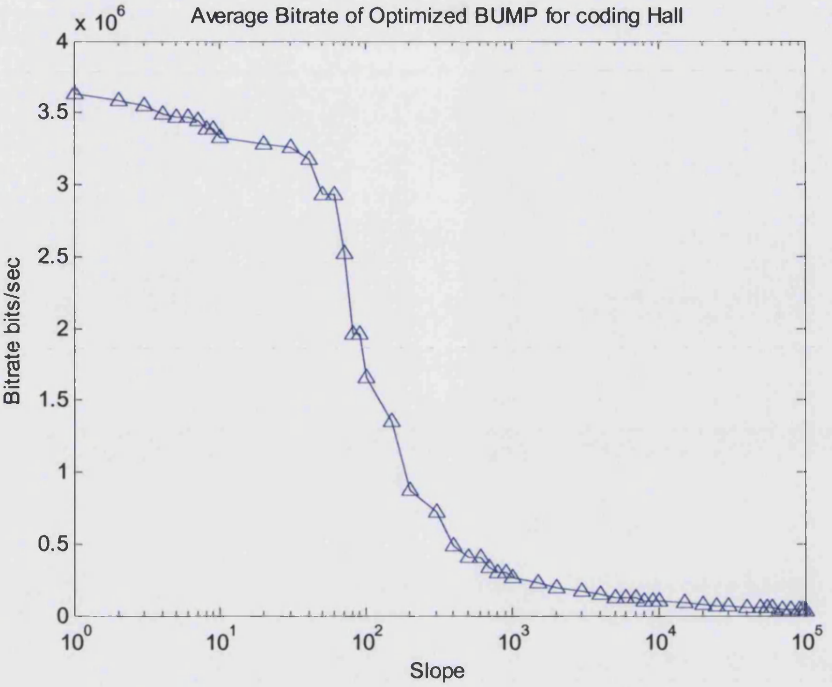


Figure 6.11 Resultant Bit rates for coding Hall

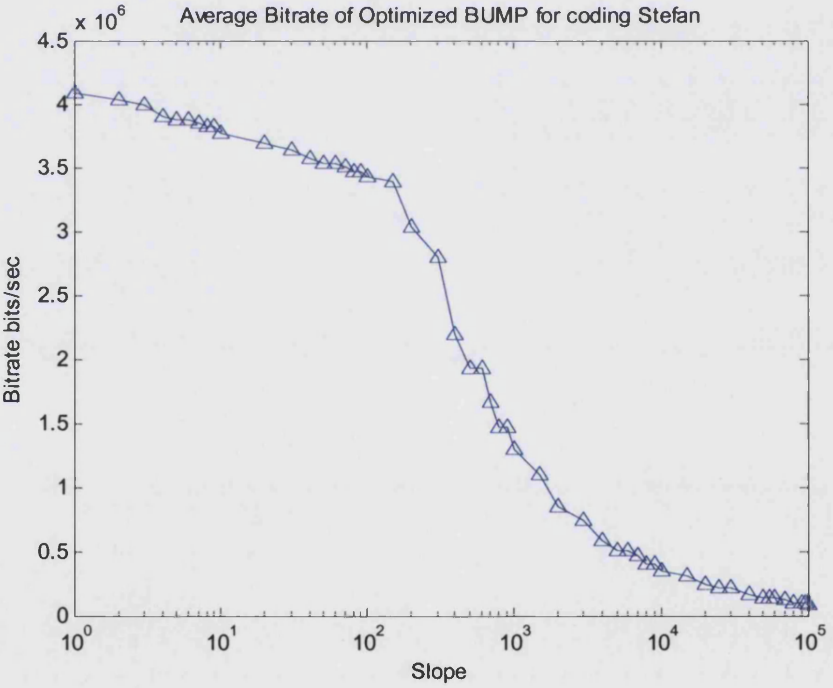


Figure 6.12 Resultant bit rates for coding Stefan

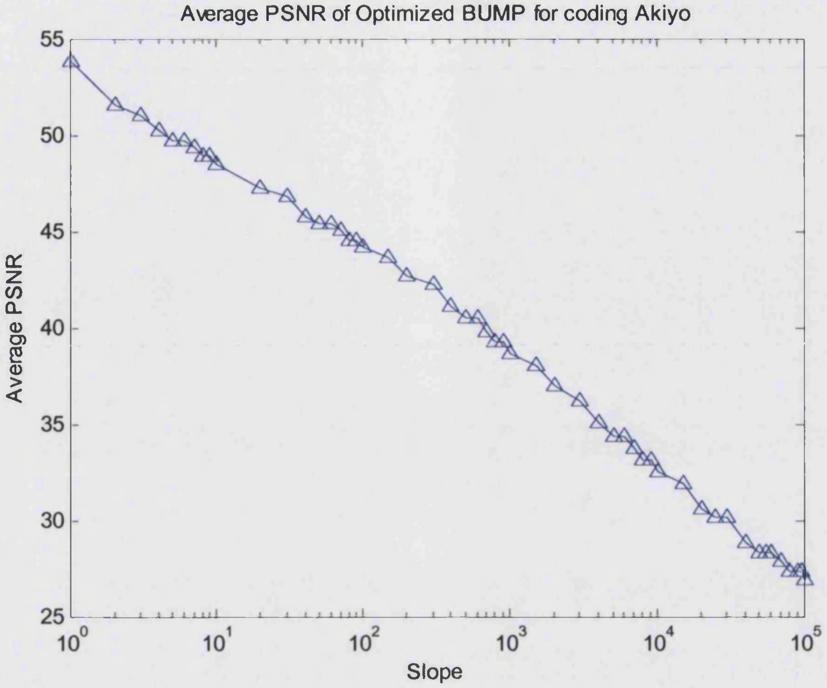


Figure 6.13 Resultant PSNR for coding Akiyo

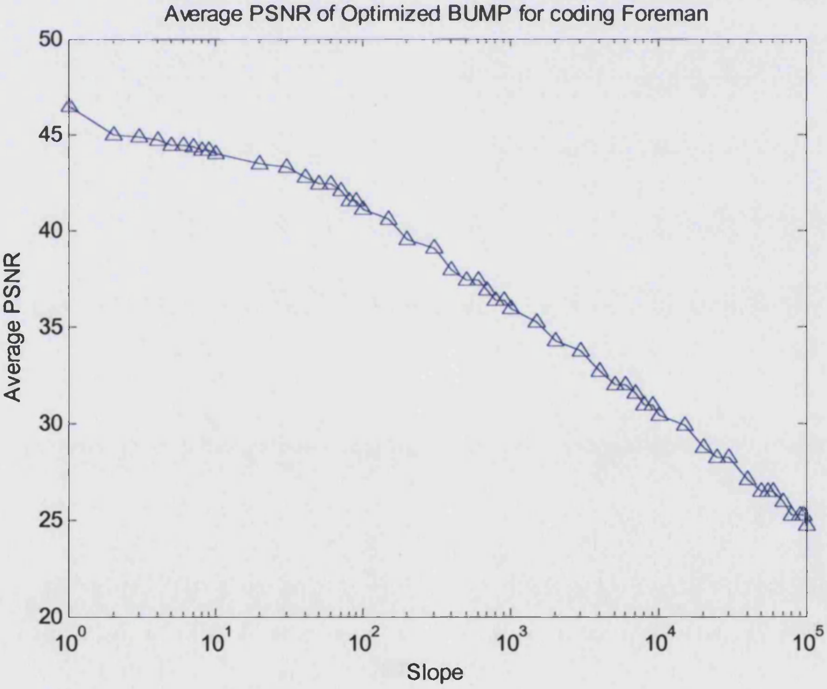


Figure 6.14 Resultant PSNR for coding Foreman

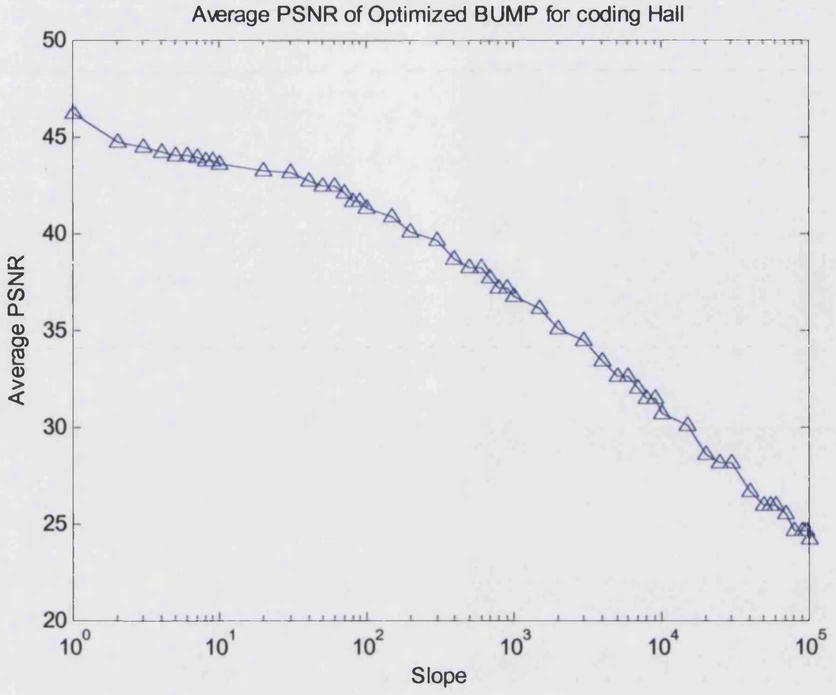


Figure 6.15 Resultant PSNR for coding Hall

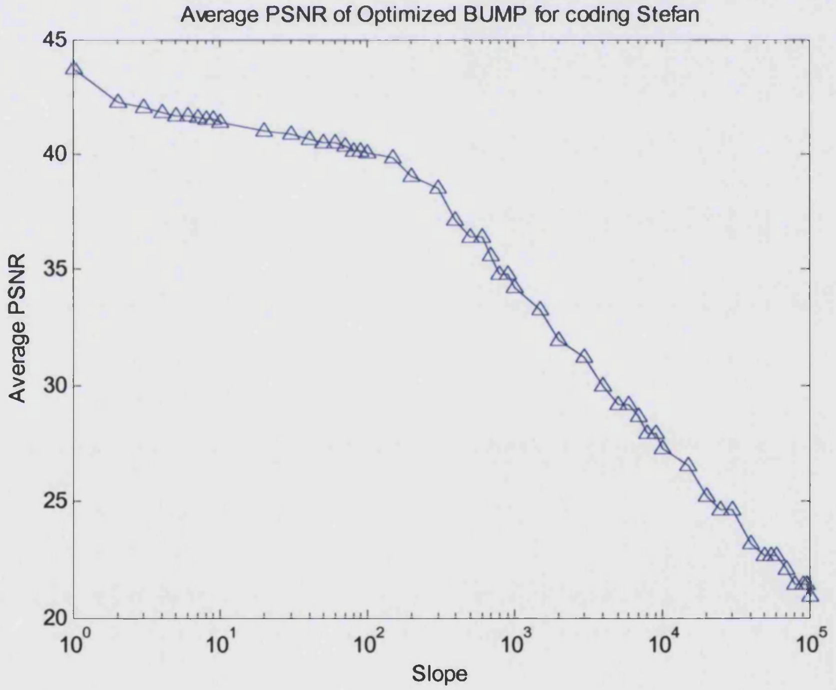


Figure 6.16 Resultant PSNR for coding Stefan

6.5.2 The rate distortion coding performance

From the Figure 6.17 to Figure 6.20, the above experimental results are plotted in such way that the relationship between the bit rate and PSNR is revealed. Again, the exact plot shape is content dependent. But it also shows some character in common. The curves all have very steep upward in the low bit rate, which indicates the scheme managed to achieve very large

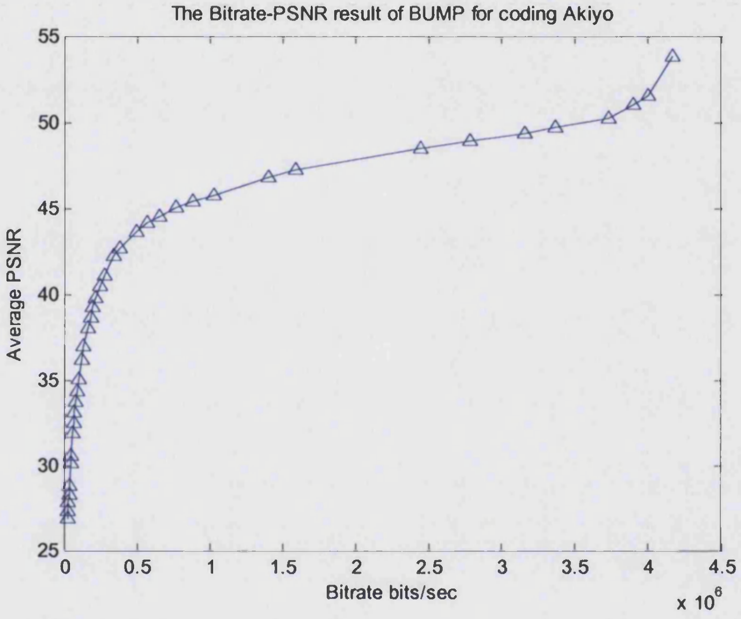


Figure 6.17 The optimized R-D performance for coding Akiyo

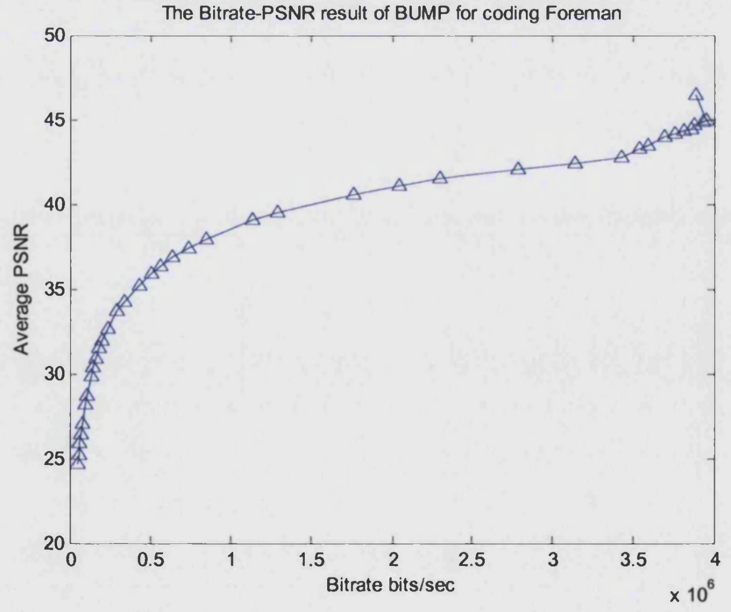


Figure 6.18 The optimized R-D performance for coding Foreman

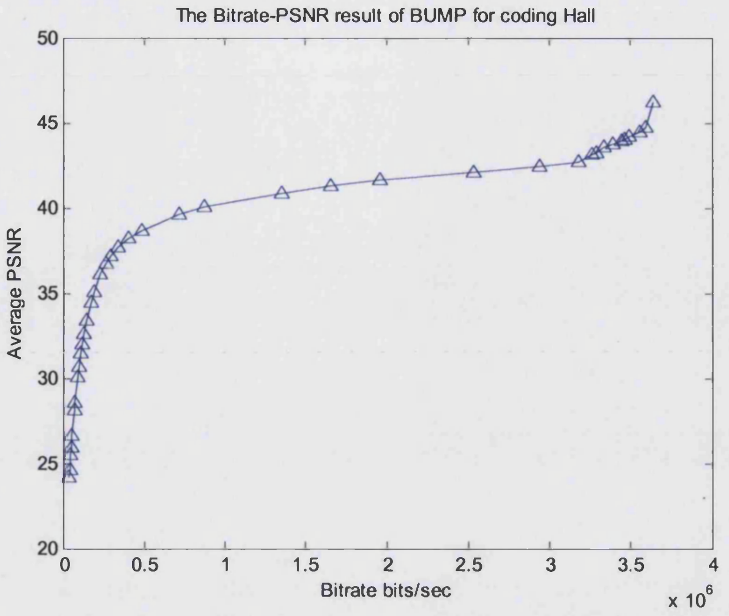


Figure 6.19 The optimized R-D performance for coding Hall

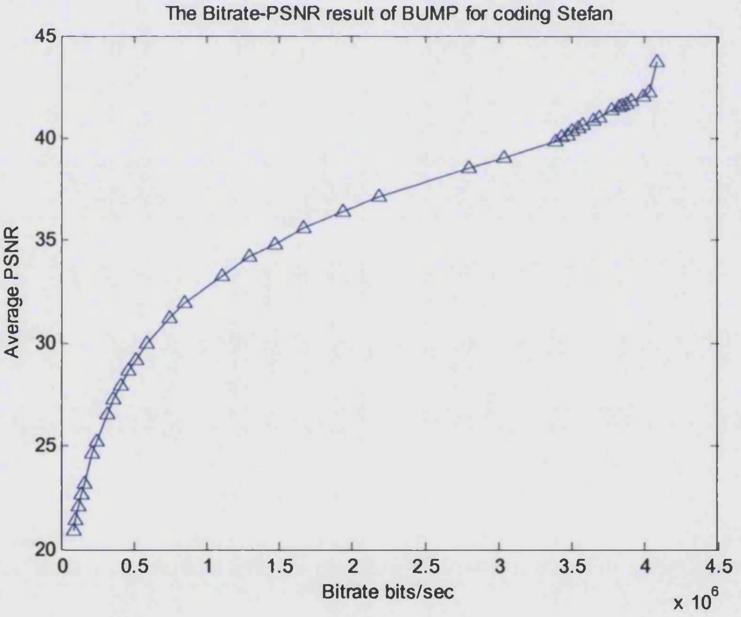


Figure 6.20 The optimized R-D performance for coding Stefan

The low bit rate part of Stefan does not show so very steep change, as in the other four. This is mainly because of the large motion content involved in the sequence, which causes the residue frame to contain large distortion, and the BUMP code does not show much advantage.

To verify the effectiveness of the proposed optimization scheme, it is compared alongside with the original BUMP coder without optimization. As showed in Chapter 4, an easy way of applying BUMP inter frame coder into video system is to build a two pass coder and let the H.264 coder decide the number of bits to use for each frame. This number is then followed by the BUMP coder. The difference between these two schemes demonstrates the benefit of applying the proposed optimization method.

The H.264 is taken as a benchmark to compare with it. In the experiments of H.264, the JM92[69] is used. The same GOP structure, as exploited in BUMP experiments, is used, and the same frames are encoded. Specifically, some parameters of H.264 are set as:

Parameter	**Value**
Entropy coding method	*CAVLC*
Rate control	*Enabled.*
Basic Unit	*1*
Initial QP	*0*
RDO feature	*Enabled*

Table 6.1 Some parameter values for experiments of H.264

The BUMP coder uses adaptive run length coding with adaptive Huffman table, so using CAVLC in H.264 is comparable. The rate control feature of H.264 is enabled, and basic unit is set as 1, which means the rate control is to be performed at MB level allowing the highest precision. It should be noted that the Initial QP for rate control is set to be 0, which means a proper Initial QP is to be selected so that the final performance is improved.

Since the target slopes value in the BUMP experiments are taken from a broad range, the results also cover an extensive range of bit rates. To make a clearer comparison with H.264, only the parts close to the bit rate space of the H.264 curve is plotted in the figures.

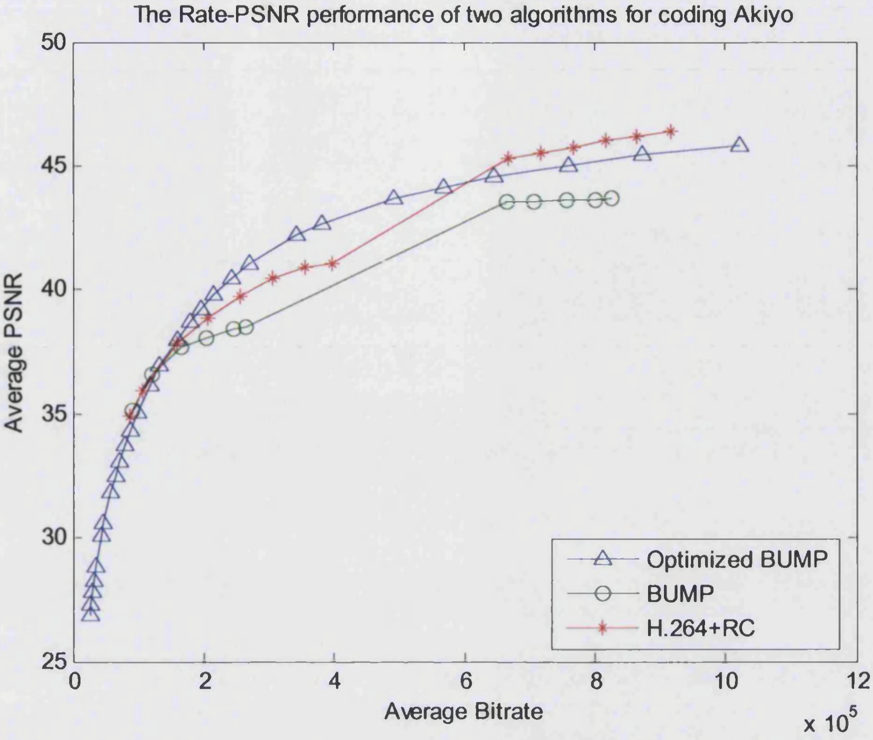


Figure 6.21 Three schemes' R-D performance for coding Akiyo

For coding Akiyo, as shown in Figure 6.21, the proposed scheme significantly improves the performance of the BUMP coder at all bit rates. Especially at low bit rate, it provides an improvement of up to 3 dB in PSNR. As a result, it even outperforms H.264 at low bit rate with a substantial advantage of near 2 dB in PSNR.

But the H.264 shows advantage again from 66kbps. Although it is not a very high bit rate, since the sequence Akiyo contains very little motion content, such bit rate is already high enough to provide a very good PSNR result. Under such condition, the residue contains very little content, and therefore BUMP coder does not show much advantage.

The result of coding Foreman is plotted in Figure 6.22. This time, the three schemes give very close performance at low bit rate. The proposed scheme shows improvement over the simple BUMP scheme at medium rate. This is a very close performance to the rate controlled H.264. Again at high bit rate, the H.264 shows advantage. It is likely that for this specific sequence, the H.264's rate control scheme is very effective, and it even brings up the performance of H.264. Such improvement is more likely to come from the proper bit allocation between the intra frame and the inter frames.

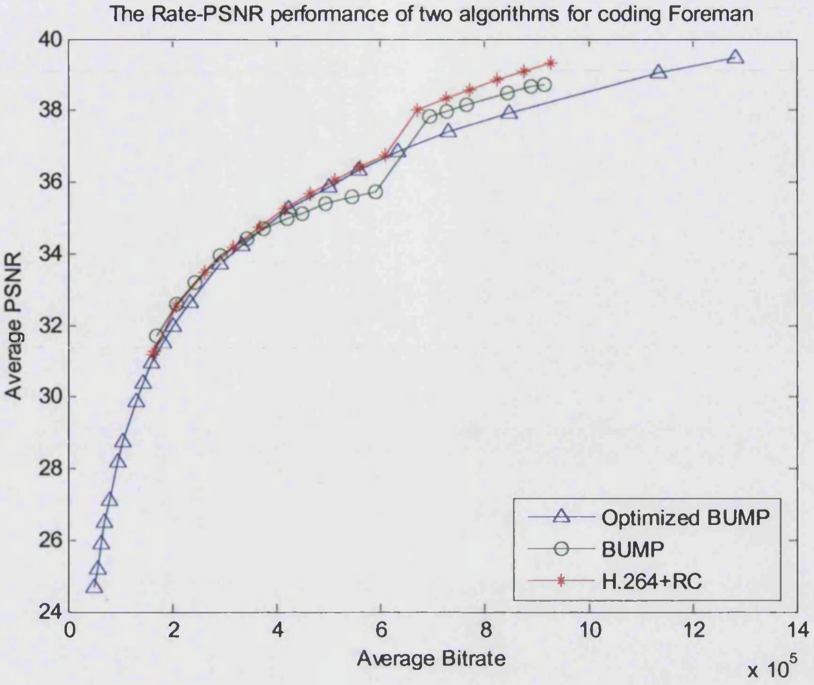


Figure 6.22 Three schemes' R-D performance for coding Foreman

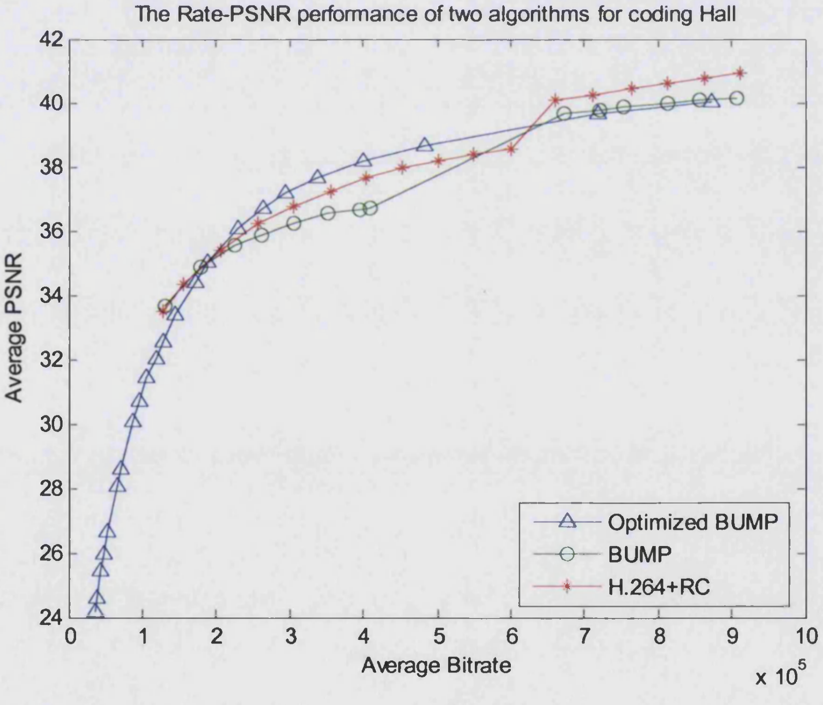


Figure 6.23 Three schemes' R-D performance for coding Hall

The result for coding Hall is plotted in Figure 6.23. Similar to Akiyo, the proposed method shows obvious improvement over original BUMP at medium rate, and it again even outperforms H.264 at low and medium rate. Finally, at high bit rate, the proposed scheme is very close to the original BUMP, and H.264 again shows the highest efficiency among the three methods.

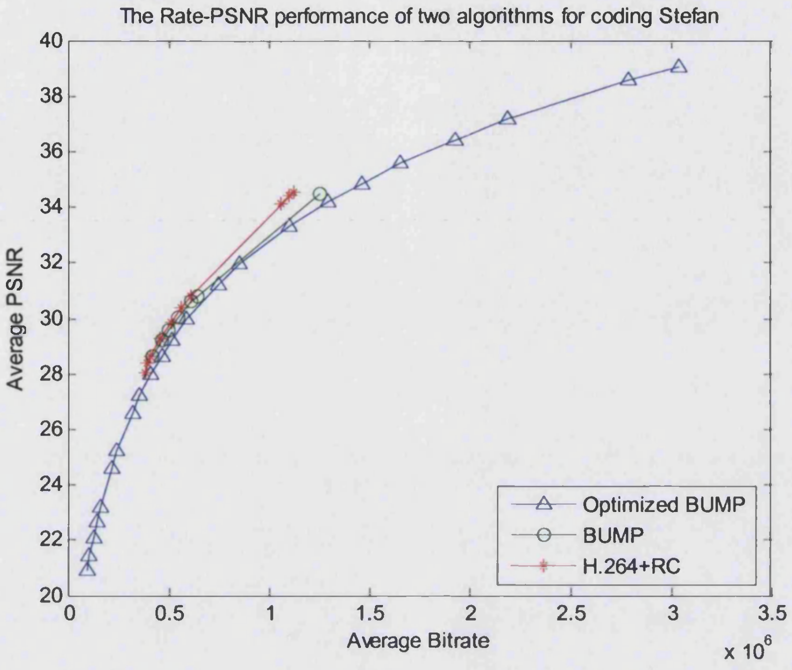


Figure 6.24 Three schemes' R-D performance for coding Stefan

For coding Stefan, the proposed scheme does not show advantage over the original BUMP. The three schemes are quite close in performance at medium rate, and H.264 shows a slight superiority at high bit rate. Because of the large amount of motion content involved, Stefan requires more bits to be encoded, comparing to the other sequences. Also, in this case, the rate control scheme of H.264 [78] cannot code the sequence at low target bit rate. But by modifying the target R-D slope, the proposed scheme is able to code the sequence at a wider range of target bit rate values.

6.6 Conclusion and discussion

In Chapter 4, the hybrid video system with H.264 and BUMP is presented, and a bits tuning method is applied. However, the method is ad hoc, and the experiments are performed on an extreme GOP structure, where one intra frame is followed by a lot of inter frames. This special structure is applied in order to compare the coding efficiency of the BUMP residue coder. In this chapter, the more general case is to be studied, and experiments are performed over a more practical GOP structure.

With the rate and distortion model proposed in Chapter 5, rate and distortion optimization can be performed to improve the BUMP coder's efficiency. The R-D slope unifying scheme fits in the BUMP coder very well and is used to for optimizing the coder. An optimization scheme for BUMP video coder is proposed. It tries to find the optimal point to cease coding by comparing the current R-D slope and the predicted future R-D slope.

Comparing to the ad hoc bit rate tuning method in Chapter 4, this scheme provides a better optimization based on novel rate and distortion models. The experiments are also performed by encoding a much more practical GOP structure. It is seen from the results that the optimized BUMP video coder provides excellent coding performance, especially in low and medium bit rate. In general, it provides a substantial improvement over the original BUMP coder at most bit rate, and as a result, it is competitive to rate controlled H.264. Among the four tested sequences, it outperforms H.264 with considerable gap in two of them at low and medium rate, and is competitive to over other range.

Chapter 7 CONCLUDING REMARKS

In this chapter, the thesis is concluded by summarizing the main contribution and some future work directions are addressed.

7.1 Conclusion

The thesis has addressed a number of aspects of the rate control problem in video coding. The objective of this thesis has been set to improve the video coder's ultimate performance by optimizing the allocation the bit budget. Both the conventional transform video coding system and the latest developed matching pursuit system are studied. According to their different mechanism, specific algorithms have been developed.

7.1.1 Rate distortion optimization for the conventional transform video coding

The state-of-the-art video standard H.264 is taken as the object of the conventional hybrid video coder. As the latest video coding standard, H.264 provides the best coding performance in literature. The new built in feature of rate distortion optimization in H.264 makes rate control even more difficult. To avoid the errors from model based estimation, an operational quantization parameter picking scheme is proposed. According to the constant rate distortion slope idea, a slope unifying scheme is proposed. Yet, since the allocation is carried out at macroblock level, the monotonic property of rate-QP relationship is not valid and that means it is difficult to carry out the slope searching.

Lagrangian multiplier is finally employed for this optimization process. This makes the choosing process very simple and straightforward. Each macroblock is taken as a stage of the coding procedure, and the candidate QP values for that stage are limited to a small number. The optimization is then converted to a problem of searching the best path. The picked candidate is chosen by Lagrangian multiplier method.

Experiments reveal that the H.264 rate control scheme makes drastic mistakes in coding some sequences and therefore has difficulty in achieving those bit rates. As a result, it also gives very poor quality in such cases. The proposed method provides a smooth rate adjustment and allows more possibility of bit rates. Also, it shows advanced efficiency on some sequences.

7.1.2 Video coding system with the H.264 motion module and matching pursuit

As another type of image coding schemes, embedded coding provides an easy approach for rate control problem and is applied to video system. However, experiments show that these techniques do not afford as high efficiency as H.264 system. The BUMP coding algorithm is then applied into the video system because it not only provides outstanding coding performance, but also enables more precise rate control. A hybrid video coder with H.264 and matching pursuit is then presented.

A simple and effective bit rate tuning method is then designed which effectively improves the final compressed video quality. It is discovered that the average coding efficiency is affected by both the Initial QP and another parameter MaxAtom. MaxAtom defines the allowed maximum number of atoms to be encoded for a single frame. Limiting the number of atoms to code within a frame might allow a smoother bit allocation along the frames in a GOP, which then may increase the general quality. By setting the two parameters, we can tune the hybrid video coder. Experiments show that it outperforms H.264 at some medium and low bit rates.

7.1.3 Bit rate and distortion modelling for matching pursuit coder

The bit rate tuning algorithm is ad hoc and requires a number of repetitions to find the optimal parameter setting. To reveal the character of the BUMP coder and develop more general applications, Chapter 5 tries to develop a source R-D model that depicts the source character so that better optimization can be implemented. According to the design of BUMP coder, the model comprises two parts, a distortion-atom model, and rate-atom model.

The coding mechanism of MP enables easy distortion measure, and the difficulty is mainly on the second part. The PLQ and MERGE coder in BUMP are studied in depth, and finally, a novel R-A model for rate estimation is proposed. The model is of very simple form and gives very high precision. An adaptive rate estimation method is then designed. Experiments prove that it is very accurate and effectively predicts the variations of the bit rate changes.

7.1.4 Rate-distortion optimization for matching pursuit video coder

With the rate and distortion models developed in Chapter 5, R-D optimization is then possible to improve the coder's effectiveness. The slope unifying method proposed in chapter 3 is difficult to be applied on macroblock basis. However, it fits in the BUMP algorithm very well. In the case of encoding a frame with MP, the R-D model enables the slope to be calculated with a single pass, and the slope of future atom can also be easily predicted. Unifying slope can then be performed by comparing the slopes and ceasing the coding when the target slope is reached. The scheme is applied to the hybrid video coding scheme with MP and H.264. The result of experiments proves its success. Comparing to the BUMP coder without the optimization scheme, it gives a considerable improvement of up to 3 dB in PSNR. It is also competitive to the H.264 with rate control. Among the tested sequences, the proposed algorithm shows an outstanding efficiency especially in low and medium bit rates, and still competitive over other bit range.

7.2 Future work

Some ideas of the future extension of the proposed methods are listed in the following,

- Real time H.264 rate control technique

  The operational algorithm for H.264 requires multi pass in picking the QP values. This is expensive in computation and is not appropriate for low delay and real time application. It is possible to simply the process by some prediction. Also, properly controlling the coding unit size reduces the times of making picking decision.

- Better source modelling for BUMP (using all the side information)

  The proposed rate estimation model for BUMP is a novel approach for rate estimation. However, it is still possible to further improve it. The MERGE coder employed in BUMP codes the atoms in a recursive fashion. There are a lot of factors that affect the final bits number. All the data about the atoms are available before the atom is coded. The model's accuracy can be further improved by considering all the available side information.

- Target rate achieving and QP choice for BUMP based video coding

  The model based optimization scheme in Chapter 6 achieves various bit rates by using different target slopes. The relationship between the target slope and the bit rate can be further studied to explore a quicker scheme to achieve the target bit rate. Also, the choice of QP values can also be adjusted to further improve the performance. First, the empirical equation suggested in the H.264 reference software may not be optimal for tuning the motion model. Second, the I frames in the system is still encoded by H.264, and properly choosing QP to control how I frames are coded can make a contribution to the over all average quality.

Author's publications

Haoxiang Zhang, X. Wang, W. Huo and D. M. Monro, "A hybrid video coder based on H.264 with matching pursuits," in *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 889-892, 2006.

Haoxiang Zhang, Adrian Evans and D. M. Monro, "A rate atom estimation model for hybrid matching pursuit video coding," to appear in Journal of Multi-media, issue 3 of 2007.

References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journey*, vol. 27, pp. 379-423, 623-656, 1948.

[2] "MPEG Test Model 5 overview," vol. 2006, 1993.

[3] I. I. 13818-2 and I.-T. R. H.262, "Generic coding of moving pictures and associated audio, Part 2: Video," 1994.

[4] M. v. group, "MPEG-4 Video Verification Model Version 8.0."

[5] I. I. F. 14496-2, "Gneric coding of audio-visual objects: (MPEG-4 video), Final Draft International Standard," in *Document N2502*, 1999.

[6] J.-C. Tsai and C.-H. Shieh, "Modified TMN8 rate control for low-delay video communications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, pp. 864, 2004.

[7] I.-T. R. H. V. 2, "Video coding for low bit rate communication," Jan 1998.

[8] G. M. Schuster and A. K. Katsaggelos, "A theory for the optimal bit allocation between displacement vector field and displaced frame difference," *Selected Areas in Communications, IEEE Journal on*, vol. 15, pp. 1739-1751, 1997.

[9] G. M. Schuster and A. K. Katsaggelos, "A video compression scheme with optimal bit allocation among segmentation, motion, and residual error," *Image Processing, IEEE Transactions on*, vol. 6, pp. 1487-1502, 1997.

[10] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers," *Multimedia, IEEE Transactions on*, vol. 1, pp. 3-17, 1999.

[11] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, pp. 182-190, 1996.

[12] J. ISO/IEC and ITU-T, "Committee Draft of Joint Video Specification (ITU-T Rec.H.264/ISO/IEC 14496-10 AVC)," in *Doc. JVT-C167*, May 2002.

[13] H. Schwarz and T. Wiegand, "An Improved H.26L Coder Using Lagrangian Coder Control, " Porto Seguro, Brasil, ITU-T/SG16/D.146,, June 2001. 2001.

[14] E. C. Reed and J. S. Lim, "Optimal multidimensional bit-rate control for video communication," *Image Processing, IEEE Transactions on*, vol. 11, pp. 873-885, 2002.

[15] I. I. 11172, "Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbps," 1993.

[16] I.-T. R. H.261, "CODEC for audio-video services at p x 64 Kbps," 1993.

[17] I.-T. R. H. V. 1, "Video coding for low bitrate communicatiuon," Mar 1996.

[18] S. Liu and C.-C. J. Kuo, "Joint Temporal & Spatial Bit Allocation for Video Coding With Dependency," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, pp. 15-26, 2005.

[19] J. Ribas-Corbera and S.-M. Lei, "A frame-layer bit allocation for H.263+," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, pp. 1154-1158, 2000.

[20] M. Jiang and N. Ling, "Bit allocation scheme for low-delay H.264/AVC rate control " in *IEEE International Conference on Image Processing,* . Atlanta, GA, USA, 2006.

[21] Z. He and S. K. Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, pp. 970-982, 2002.

[22] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding: a summary," *Circuits and Systems Magazine, IEEE*, vol. 2, pp. 46-49, 2002.

[23] Z. He and S. K. Mitra, "Optimum bit allocation and accurate rate control for video coding via /spl rho/-domain source modeling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, pp. 840-849, 2002.

[24] Z. He and T. Chen, "Linear rate control for JVT video coding," presented at Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on, 2003.

[25] "Coding of Audiovisual Objects—Part 10: Advanced Video Coding," Advanced video coding for generic audiovisual services, ISO/IEC 14496-10:2003, ITU-T Recommendation H.264,," 2003.

[26] Iain.E.G.Richardson, *H.264 and MPEG-4 Video Compression.* Chinchester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd, 2003.

[27] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 560-576, 2003.

[28] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, pp. 172-185, 1999.

[29] Z. G. Li, N. Ling, G. N. Feng, F. Pan, K. P. Lim, and S. Wu, "Adaptive rate control for real time video coding process," presented at Digital and Computational Video, 2002. DCV 2002. Proceedings. Third International Workshop on, 2002.

[30] S. Ma, W. Gao, P. Gao, and Y. Lu, "Rate control for advance video coding (AVC) standard," presented at Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on, 2003.

[31] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 41, pp. 3445-3462, 1993.

[32] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, "CREW: Compression with Reversible Embedded Wavelets," presented at Data Compression Conference, 1995. DCC '95. Proceedings, 1995.

[33] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, pp. 243-250, 1996.

[34] J.L. Mitchell, W.B. Pennebaker, C.E.Fogg, and D. J. Legall, *MPEG: Video Compression Standard,*. New York and London: Chapman and Hall, 1996.

[35] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*. London: IEE, 2003.

[36] J. Watkinson, *Compression in Video and Audio*. Oxford England: Focal Press, 1995.

[37] T. Sikora, "Trends and perspectives in image and video coding," *Proceedings of the IEEE*, vol. 93, pp. 6-17, 2005.

[38] G. K. Wallace, "JPEG: Still image compression standard," *Communications of the ACM*, vol. 34, pp. 30-44, 1991.

[39] N. Ahmed, T. Natrajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Transaction On Computing*, vol. C-23, pp. 90-93, 1984.

[40] S. G. Mallat, *A wavelet tour of signal processing*: Academic Press, 1998.

[41] T. N. Gilbert Strang, *Wavelets and Filter Banks,*, 2nd ed. Wellesley MA,: Wellesley-Cambridge Press, 1997.

[42] "ITU Telecom. Standardization Sector of ITU "Video coding for low bitrate communication," ITU-T Recommendation H.263," 1996.

[43] J. L. Mitchell, W.B. Pennebaker, C.E.Fogg, and D. J. Legall, *MPEG: Video Compression Standard,* New York and London: Chapman and Hall, 1996.

[44] K. Rijkse, "H.263: video coding for low-bit-rate communication," *Communications Magazine, IEEE*, vol. 34, pp. 42-45, 1996.

[45] "ITU Telecom. Standardization Sector of ITU "Video coding for low bitrate communication," Draft ITU-T Recommendation H.263 Version 2," in *Draft ITU-T Recommendation H.263 Version 2,* 1997.

[46] T. Sikora, "MPEG digital video-coding standards," *Signal Processing Magazine, IEEE*, vol. 14, pp. 82-100, 1997.

[47] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: video coding at low bit rates," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, pp. 849-866, 1998.

[48] S. Battista, F. Casalino, and C. Lande, "MPEG-4: a multimedia standard for the third millennium. 1," *Multimedia, IEEE*, vol. 6, pp. 74-83, 1999.

[49] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *Signal Processing Magazine, IEEE*, vol. 18, pp. 36-58, 2001.

[50] A. Tamhankar and K. R. Rao, "An overview of H.264/MPEG-4 Part 10," presented at Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on, 2003.

[51] G. K. Wallace, "The JPEG still picture compression standard," *Consumer Electronics, IEEE Transactions on*, vol. 38, pp. xviii-xxxiv, 1992.

[52] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," *Proceedings of the IEEE*, vol. 83, pp. 151-157, 1995.

[53] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *Image Processing, IEEE Transactions on*, vol. 1, pp. 205-220, 1992.

[54] D. Taubman, "High performance scalable image compression with EBCOT," *Image Processing, IEEE Transactions on*, vol. 9, pp. 1158-1170, 2000.

[55] I. I. FCD15444-1, "JPEG2000 Final committee draft version 1.0," Mar 2000.

[56] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *Image Processing, IEEE Transactions on*, vol. 6, pp. 677-693, 1997.

[57] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Wavelet packet image coding using space-frequency quantization," *Image Processing, IEEE Transactions on*, vol. 7, pp. 892-898, 1998.

[58] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *Information Theory, IEEE Transactions on*, vol. 36, pp. 961-1005, 1990.

[59] Z. Xiong and X. Wu, "Wavelet image coding using trellis coded space-frequency quantization," *Signal Processing Letters, IEEE*, vol. 6, pp. 158-161, 1999.

[60] D. A. Huffman, "A method for the construction of minimum redundancy codes," *In Proceedings IRE*, vol. vol. 40,, pp. pp. 1098-1101, 1962.

[61] G. G. Langdon, "An Introduction to Arithmetic Coding," *IBM Journal of Research and Development*, vol. 28, pp. 135-149, 1984.

[62] Wikipedia, "MJPEG," Wikipedia, 2006.

[63] J. Jain and A. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *Communications, IEEE Transactions on [legacy, pre - 1988]*, vol. 29, pp. 1799, 1981.

[64] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, pp. 74-90, 1998.

[65] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers [speech coding]," *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 36, pp. 1445, 1988.

[66] L. Cooper, *Introduction to dynamic programming*: Pergamon, 1981.

[67] P. Whittle, *Optimization over time : dynamic programming and stochastic control. Vol.2.* . Chichester Wiley, 1983.

[68] J. Ribas-Corbera, P. A. Chou, and S. L. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 674-687, 2003.

[69] K. Sühring, "H.264/AVC Reference Software," vol. 2005, 2005.

[70] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia* Chinchester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd, 2003.

[71] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 688-703, 2003.

[72] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 598-603, 2003.

[73] M. Wien, "Variable block-size transforms for H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 604-613, 2003.

[74] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 620-636, 2003.

[75] P. List, A. Joch, J. Lainema, G. Bjntegaard, and M. Karczewicz, "Adaptive deblocking filter," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 614-619, 2003.

[76] P. Noll, "MPEG digital audio coding," *Signal Processing Magazine, IEEE*, vol. 14, pp. 59-81, 1997.

[77] Satoshi Miyaji, Yasuhiro Takishima, and Y. Hatori, "A Novel Rate Control Method for H.264 Video Coding," presented at International Conference on Image Processing, 2005.

[78] S. Ma, W. Gao, P. Gao, and Y. Lu, "Rate control for advance video coding (AVC) standard," 2003.

[79] S. Ma, W. Gao, F. Wu, and Y. Lu, "Rate control for JVT video coding scheme with HRD considerations," presented at IEEE International Conference for Image Processing, 2003.

[80] G. C. K. Abhayaratne, "Lossless and Near Lossless Digital Video Coding," PhD Thesis, University of Bath, UK, 2002.

[81] N. K. Laurance and D. M. Monro, "Embedded DCT coding with significance masking," presented at Int. Conf. On ASSP, 1997.

[82] C. Tu and T. D. Tran, "Context-based entropy coding of block transform coefficients for image compression," *Image Processing, IEEE Transactions on*, vol. 11, pp. 1271, 2002.

[83] W. Poh and D. M. Monro, "Improved compression of motion-compensated residuals," presented at IEEE Int.Conf. on Image, 2002.

[84] Z. Xiong, O. G. Guleryuz, and M. T. Orchard, "A DCT-based embedded image coder," *Signal Processing Letters, IEEE*, vol. 3, pp. 289-290, 1996.

[85] S. Golomb, "Run-length encodings (Corresp.)," *Information Theory, IEEE Transactions on*, vol. 12, pp. 399, 1966.

[86] G. G. Langdon, "An adaptive run-length coding algorithm," *IBM Technical Disclosure Bulletin,* , vol. 26, pp. 3783-3785, 1983.

[87] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397-3415, 1993.

[88] R. Neff and A. Zakhor, "Very low bit-rate video coding based on matching pursuits," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, pp. 158-171, 1997.

[89] Y. Yuan and D. M. Monro, "Improved Matching Pursuits Image Coding," presented at Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, 2005.

[90] Y. Yuan, A. N. Evans, and D. M. Monro, "Low complexity Separable Matching Pursuits," presented at IEEE Int. Conf. Acoustics Speech Signal Process. (ICASSP 2004), Montreal, 2004.

[91] D. M. Monro, "Basis Picking for Matching Pursuits Image Coding," presented at IEEE Int. Conf. Image Process. (ICIP2004), Singapore, 2004.

[92] H. Zhang, X. Wang, W. Huo, and D. M. Monro, "A Hybrid Video Coder Based on H.264 with Matching Pursuits," presented at IEEE International Conference on Acoustics, Speech and Signal Processing, Toulouse, France, 2006.

[93] D. M. Monro, J.-L. Aufranc, M. A. Bowers, and W. Poh, "Visual embedding of wavelet transform coefficients," presented at IEEE Int. Conf. Image Process. (ICIP 2000), Sept. 2000.

[94] W. Poh and D. M. Monro, "Comparison of residual compression methods in motion compensated video," presented at IEEE Int. Workshop on Multimedia Signal Processing,, Virgin Islands,, 2002.

[95] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *Circuits and Systems Magazine, IEEE*, vol. 4, pp. 7-28, 2004.

[96] A. Ortego and K. Ramchandran, "Rate-distortion methods for image and video compression," *Signal Processing Magazine, IEEE*, vol. 15, pp. 23-50, 1998.

[97] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, pp. 446, 1998.

[98] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *Image Processing, IEEE Transactions on*, vol. 3, pp. 533-545, 1994.

[99] Z. Li, F. Pan, K. P. Lim, G. Feng, X. Lin, and S. Rahardja, "Adaptive Basic Unit layer Rate Control for JVT," Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, Pattaya, Thailand, Proposal 7-14 March 2003.

[100] R. Neff and A. Zakhor, "Matching-pursuit video coding .II. Operational models for rate and distortion," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, pp. 27-39, 2002.

[101] Z. He and S. K. Mitra, "From Rate-Distortion Analysis to Resource-Distortion Analysis," *IEEE Circuits and Systems Magazine*, pp. 6-18, 2005.

[102] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, pp. 246-250, 1997.

[103] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 6, pp. 12-20, 1996.