This is to certify that the

thesis entitled

RATE DISTORTION BASED RATE CONTROL
METHODS FOR SCALABLE VIDEO

presented by

SHARADHA PARTHASARATHY

has been accepted towards fulfillment
of the requirements for

M.S. degree in ELECTRICAL ENGR.

Major professor

Date 7/23/02

RATE-DISTORTION BASED RATE-CONTROL METHODS FOR SCALABLE VIDEO

By

Sharadha Parthasarathy

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

2002

ABSTRACT

RATE-DISTORTION BASED RATE-CONTROL METHODS FOR SCALABLE VIDEO

By

Sharadha Parthasarathy

Scalable video coding techniques are used to facilitate the delivery of streamed video content over networks with varying channel characteristics and unpredictable quality-of-service (QoS) such as the Internet and wireless Local-Area-Networks (LANs). Rate control methods are typically used with scalability solutions to adapt the video to variations in bandwidth availability. In this thesis, we develop two rate- distortion (RD) based rate-control algorithms for scalable video coded using the new fine-grained scalability (FGS) technique incorporated in the MPEG-4 video-coding standard. First, we show that the distortion measures, which are based on a square error criterion, can be computed using a very simple approach in the transform (DCT) domain. Second, we describe the two RD algorithms and their implementations within the transform domain of the FGS MPEG-4 framework. Finally, we present the results of experiments conducted to test the performance of the RD-based algorithms against the existing rate-control method. We find, upon analysis of these results in the transform domain, that the proposed rate-control algorithms provide optimum (convex) RD curves for a wide range of bitrates and video sequences. Improvements shown in the RD curves are particularly salient over bitrate ranges that can be supported over emerging Internet access technologies (e.g., DSL, cable modems, and wireless LANs).

To My Parents and Meera,
For all your love and support

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Hayder Radha, for providing me with the opportunity to work with him. Without his guidance and support over the last two years, this thesis would not be possible. I would also like to thank the members of my defense committee, Dr. John R. Deller and Dr. Percy Pierre, for their valuable comments and advice.

Thanks are also due to all the members of the WAVES lab at Michigan State University for their constructive reviews of my work and more importantly for their friendship and support. I would specially like to thank Aparna Gurijala for always being there for me.

Finally, I would like to thank my parents and my sister for teaching me the importance of hard work and patience, for their perpetual belief in my abilities and for their irreplaceable support through all these years.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

The advent of the World Wide Web has essentially transformed the Internet into the world's largest public network. The popular applications have evolved from being file transfer and e-mail to video conferencing, Internet telephony, Web searching and multimedia streaming. For streaming applications in particular, a primary consideration is the necessary ability to transmit data at sustained and relatively high rates. Recent technological advances enable the network to handle these data rates. Some examples of the new tools proposed in this regard include scalability and rate control.

In this chapter, we first state the objective of the work presented in this thesis. We then provide an introduction to streaming video over the Internet and briefly examine certain constraints posed by this network on such an application. We also discuss the scalability techniques used to work around these constraints. We conclude this chapter with an overview of the organization of the rest of this thesis.

1.1 Objective Of Thesis Work

One of the scalability techniques proposed for the MPEG-4 video-coding standard is *fine-granularity scalability* (FGS)[17]. This technique involves coding the video into two layers- a *base* layer and a single *enhancement* layer. The base-layer is the non-scalable video component. It determines the minimum quality of the reconstructed video at the receiving

end. The enhancement layer is the scalable video component. In MPEG-4 FGS, the enhancement layer is partially transmitted and its size depends on the available bandwidth for transmitting the video. Rate-control is essentially carried out in the FGS enhancement layer by means of a brute-force raster-scan bit allocation method. Such a rate-control method has very low complexity but it is not optimal. In this thesis, we develop and analyze the performance of two rate-control algorithms that result in optimal rate-distortion performance for the FGS enhancement layer. The algorithms are operational rate-distortion algorithms, in the sense that they are designed specifically for the MPEG-4 FGS encoder-decoder system. The analysis and development presented in this thesis is carried out in the transform domain. The distortion measures selected for the two algorithms are designed to minimally increase the computational complexity. The presented RD-based algorithms are easy to implement and have low complexity. They are also independent of the rate-control scheme used in the base layer.

1.2 Streaming Video Over The Internet

The Internet has its roots in the ARPANET[26]. It was initially designed for networking research and for scientists to exchange information amongst themselves. The primary applications were download-based, wherein the transfer and viewing are temporally sequential. *Real-time* multimedia applications differ from the traditional Internet applications in that *viewing* takes place at the same time as downloading. For example, in the case of streaming applications, such as video streaming, the user views the video as the transfer occurs. In other words, transfer and viewing are concurrent. The best effort model on which

the Internet is based[41], does not offer data delivery reliability. Hence, there is a variation in the probability of packet loss. This could lead to an unacceptable quality in a video streaming application. Another concern in the streaming of video over the Internet is high bandwidth variation. The heterogeneous nature of the access-technologies of the receivers (analog modems, cable modems, DSL etc.) and congestion over the core network are two of the primary reasons for non-constant bandwidth availability.

It is necessary to overcome the problems posed by unreliable packet transmission and bandwidth availability for dependable video streaming quality. The objective of many contemporary research efforts in this field has been to find Internet video coding solutions so as to make Internet video quality comparable to television telecasts.

One generic framework that addresses the primary concerns of video coding and networking is *scalability*. A video application that is to be streamed over the Internet must "adapt" to changes in the network conditions- specifically to the variations in the available bandwidth. This is made possible by using scalable video. Scalable video coding primarily aims to provide minimum real-time processing of the video, high adaptability of the coded video to network conditions, low complexity decoding and resilience to packet-losses.

1.3 Scalability Techniques

Scalability is one of the most desirable properties in a flexible video encoder-decoder module. The basic idea here is to use *layers* of video. The layers comprise of a single non-scalable *base layer* and one or more *enhancement layers*. If a user is incapable or unwilling

to reconstruct the video at its complete resolution, scalability allows that user to decode subsets of the layered bitstream. Therefore, he or she can display the video at a lower spatial or temporal resolution or with lower quality. Another important purpose for layering of video is that it makes the bitstream suitable for prioritized transmission.

Some common tools used in recent video coding standards (such as MPEG-2) are: data partitioning, SNR scalability, spatial scalability and temporal scalability[34][28]. Hybrid scalability is also supported. Briefly, the salient features of these scalable techniques are:

Data Partitioning: Here, the bitstream is partitioned into important error-prone data (e.g., header information and motion vectors) and less-important data (e.g., transform coefficients). This method, which allows for different protection levels during transmission for the data types of data in the compressed video bitstream, has low implementation complexity. It helps in concealment of transmission errors and channel errors.

SNR Scalability: In this method, video layers at the same spatial resolution are generated. These layers, however, have different video qualities. The lower layer (usually known as the *base layer*) provides a basic video quality and the enhancement layers, when added to the base layer, augment the video quality. The layers are generated from the same source.

Spatial Scalability: This scalability technique involves creating different video layers having different spatial resolution. The layers are again, generated from a single video source. The base layer provides the basic spatial resolution. The enhancement layer(s) encode the interpolated video from the base layer.

Temporal Scalability: Here, the base layer has the basic temporal resolution. The enhancement layer(s) are coded using temporal prediction with respect to the base layer. At the decoder, the layers are temporally multiplexed to yield the full temporal resolution of the video source sequence.

1.3.1 Fine Granularity Scalability (FGS)

In the layered coding schemes mentioned above, the enhancement layer is similar to the base layer in the sense that, by itself, it is non-scalable. In other words, if the enhancement layer is not entirely received, it cannot be used to enhance the video quality at all. The desired objective is to have a video coding scheme that provides scalability progressively through the enhancement layer.

A new scalability technique called *fine-granularity-scalability* (FGS) was proposed in [17] that addressed the said objective. This technique has been included in the MPEG-4 standard for video coding[21][18][42]. The major difference between this technique and the other scalable coding techniques is that the enhancement layer can be truncated into any specified number of bits within every frame. This allows for partial enhancement proportional to the number of bits decoded for each frame.

The FGS base and enhancement layers employ a video compression algorithm wherein the original video frames are first transform coded using the *discrete cosine transform* (DCT). The DCT coefficients are then quantized and run-length encoded. The base layer is coded at a constant bitrate by means of a rate-control algorithm. A rate control algorithm is fundamentally a bit allocation algorithm, which guarantees that the *average bitrate* of the

compressed video is a coded at a *desired* bitrate. This enables any video transmitter (e.g., an Internet video server) to send the compressed bitstream at that constant rate. One rate-control mechanism that is used for the FGS base layer is the MPEG2-TM5 (test model 5) algorithm[39].

In the base layer, the DCT coefficients are first subject to quantization. The quantized values are subsequently run-length coded to form the base layer bitstream. For every frame of the video sequence, the difference between the original frame and the frame reconstructed after inverse quantization gives what is called the residual frame. The residual frame is also DCT coded. The residual DCT coefficient for every pixel of every frame is entropy encoded to form the FGS enhancement layer. The entropy coding method used for the FGS enhancement layer is bitplane coding. In this method, each residual DCT coefficient value is considered to be an integer that consists of several bits. Therefore, the residual DCT coefficients, which belong to a given FGS enhancement-layer frame, form multiple *bitplanes*. These DCT bitplanes, which range from the most-significant-bit (MSB) plane to the least-significant one, give the desired SNR (quality) scalability of the FGS structure. Details of this coding method are explained later in Chapter 3.

FGS enables the separation between the encoding process and the transmission process. In other words, the encoder compresses the bitstream prior to the time when the stream is sent to the receiver(s). This separation is a required feature for applications that rely on streaming pre-compressed video that is already stored in a web-server. When the video has to be streamed, the base layer is sent through completely using the constant bitrate that it has been coded at. An FGS-compatible streaming server sends the appropriate portion of the

enhancement layer in a raster-scan order, starting from the most-significant-bit (MSB) plane of that layer. When the server reaches the bandwidth limit (that is available between the transmitter and the receiver), it discards the remainder of the enhancement layer. This process represents a rate-control procedure that is very simple to implement, but is not optimal. In this thesis, we develop and analyze the performance of an optimal rate-distortion (RD) based rate-control mechanism for the FGS enhancement layer. Based on this framework we present two RD-based algorithms. These algorithms are easy to implement and have low complexity. They are also independent of the rate-control scheme used in the base layer.

1.4 Summary And Thesis Organization

Internet content developers have increasingly used multimedia to make information more interactive and accessible. There are a growing number of web sites that make available the option of streaming applications, particularly streaming of video content. Considerable research has been done towards improving the quality of streamed video [2][3][17][37][41][34]. Scalable video coding has been adopted to improve the overall performance and quality of the video received by the end user. Fine-Granularity Scalability (FGS) is used with the MPEG-4 video-coding standard for this purpose.

Simple rate control algorithms have been defined and developed for the FGS base layer. The enhancement layer is cut using a coarse raster-scan approach. In this thesis, we introduce a rate-distortion based solution for optimal rate-control of the FGS enhancement layer.

The remainder of this document is organized as follows. Chapter 2 provides a short discussion of classical rate-distortion theory. It also explicates the need for image data compression and lists some common encoding techniques. This is followed by deliberations on the issue of determining a suitable measure of distortion. A few important aspects of the MPEG video-coding framework are emphasized. Also, an overview of the Fine-Granularity-Scalability technique used in the MPEG-4 video-coding standard is presented in this chapter. This includes a discussion of the bitplane-coding method used in FGS and some important features of the FGS encoder and decoder. In Chapter 3, the mathematics and logistics for the bitplane rate-distortion based rate-control (BPRD) algorithm are described in detail. These are accompanied by a comprehensive discussion of experimental results. In Chapter 4, the mathematical formulation, the algorithm and the results of experimental analysis of an extension to the BPRD, called the cross-bitplane rate-distortion rate-control (CBPRD) algorithm, are elucidated. The final chapter provides a summary of the conclusions made from the analysis of the two algorithms proposed in this thesis and states the possible future work in this area.

2 Background

The need for effective and standardized image compression procedures has been amplified by the rapid growth in digital imaging applications such as multimedia, teleconferencing and high-definition television (HDTV). In the past few decades, standards have emerged for still images (JPEG) and motion video (MPEG). In the case of transmitting video over a network such as the Internet, a primary concern is compressing the video to fit within the available bandwidth. A compression scheme typically introduces some error into the video. It is essential that an optimal balance be struck between the rate to which the video is restricted and the distortion introduced by such a restriction. A branch of classical information theory, called *rate-distortion* theory addresses this optimization problem.

The first part of this chapter contains a simple explanation of rate-distortion theory and its relation to the compression of image data. Some of the techniques used for still image compression and video coding are discussed. Also, the problem of determining a suitable distortion measure for different types of image data is studied.

In the second part of this chapter, a few important aspects of the MPEG video-coding framework in general and the MPEG-4 standard in particular are emphasized. MPEG-4 video is intended to provide standardized technology for efficient storage, transmission and manipulation of video data in the broad spectrum of multimedia applications. Instead of being application specific, the solutions are presented in the form of tools and algorithms

that have functionalities common to a set of applications. Efficient compression, object scalability, spatial and temporal scalability and error resilience are examples of such functionalities. Our focus in the later sections of this chapter is specifically the fine-granularity scalability (FGS) scheme included in MPEG-4.

2.1 Rate Distortion Theory

Rate-Distortion theory can be traced back to Shannon [5][6]. In his Coding Theorem, Shannon states that a source with an entropy rate of H can be transmitted reliably over any channel of capacity C subject to the condition that C>H. The converse of this theorem states that not all the source data can be recovered reliably when H>C. It follows from this that in order to recover all data correctly, we can either decrease H or increase C. We typically assume that the channels are such that they have the minimum possible H and the maximum possible C but the condition of greater entropy (H>C) persists. In such a situation, we attempt to preserve those aspects of the source data that are most essential. In other words, we attempt to minimize the *distortion* subject to the condition that the rate of transmission cannot exceed the channel capacity. This leads to a trade-off between the rate at which information is provided at the source output and the fidelity with which this output can be reconstructed at the receiving end, on the basis of the information provided. This trade-off is mathematically modeled in the context of information theory by a discipline known as Rate-Distortion theory [35]. Thus, rate-distortion theory is concerned with maximally reducing the redundancy from a source (or the number of bits necessary to represent the source

information) for a given reproduction quality. A graphical representation of a typical rate-distortion trade-off is shown in Figure 2-1

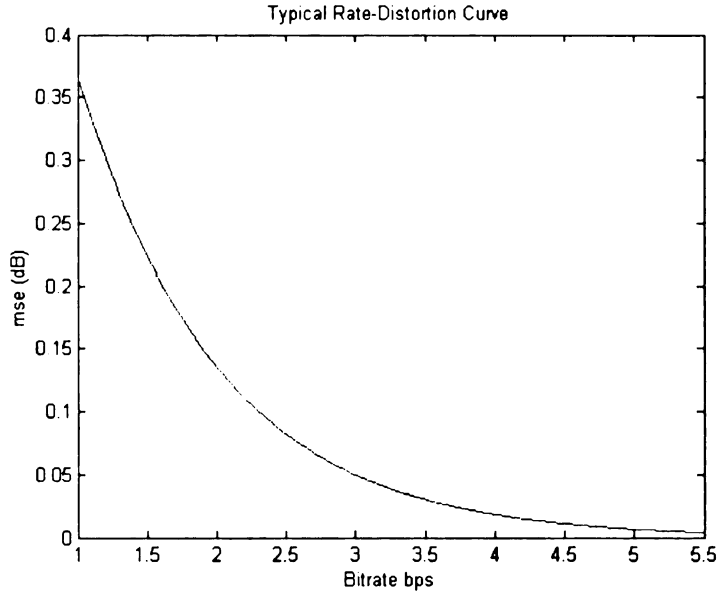


Figure 2-1 Typical (optimal) RD tradeoff curve

If the bitrate R at which information is provided exceeds the entropy H, then no information loss occurs and hence no distortion. The minimum distortion that can be obtained at any rate between R=H and R=0 increases from D=0 to some value $D=D_{max}$. If no information is transmitted (i.e., when R=0), the receiver can estimate the source by using its mean value. In this case, the mean-square distortion $D=D_{max}$ is the variance of the source. This distortion is the minimum possible value that can be made in the total absence of information about the output from the source. For a given available rate R_{max}, the objective

of an RD-based algorithm is to identify a subset $\hat{X}^*$ of the source data X that can be represented with R (or fewer) bits while minimizing some distortion measure D:

$$\hat{X}^* = \arg_{\hat{X}} \left\{ \min_{R \le R_{\max}} D(R; \hat{X}) \right\} \tag{2.1}$$

We now proceed to look at the concept of rate-distortion in the context of coding image and video signals.

2.2 Image And Video Compression

The volume of data associated with visual information is extremely high. The very conversion of an analog video signal into a digital signal calls for a tremendous increase in required transmission bandwidth. For instance, a 4 MHz television signal sampled at Nyquist rate with 8 bits per sample would require a bandwidth of 32 MHz when transmitted using a digital modulation scheme like phase-shift keying (PSK). Thus, the analog to digital conversion results in almost an eightfold increase in bandwidth [1]. Also, video image sources generate data at very high rates. Typical motion data images like television images are generated at rates greater than 10 million bytes per second. Other sources produce images at even higher bit rates. Evidently, transmission of such data requires large channel capacity, which can be very expensive and not available to all users in many applications like streaming video over the Internet. There is, therefore, a need to reduce the data volume and accommodate the data within the available bandwidth. Image data compression

techniques are applied to compensate for the increase in bandwidth caused by digital conversion of the video signal. They do so by reducing the number of bits required to transmit such data with tolerable loss of information.

2.2.1 Compression Techniques

Compression can be achieved by the use of "lossless" techniques. Here, the data obtained after decompression are an *exact* copy of the original (e.g. staple software tools like *zip* or *compress*). Lossless compression schemes can be classified as spatial coherence based encoding (e.g. run-length encoding and statistical encoding), entropy encoding (e.g. Huffman encoding, arithmetic encoding) and codebook encoding (e.g. LZW coding). Details on these compression schemes are in [27][24]. Such techniques are independent of the type of information. They are only concerned with how the information is represented. They are applicable in cases where a perfect reconstruction of the source is an essential requirement. Lossless compression techniques are mainly used for compressing digital manuscripts etc.

When image or video data are being handled, the large volume of the data, the bandwidth and storage requirements limit the performance of a lossless compression method. In such a scenario, "lossy" compression techniques are a preferred alternative [43].

Compression is said to be lossy if the decoded image is not an *exact* replica but is instead an *almost indistinguishable* reconstruction of the original image (we can recreate an image that is "indistinguishable" from the original if the properties of the human visual system are correctly exploited). Here, source fidelity is compromised for a better and more feasible coding rate. This compromise or trade-off is exactly the rate-distortion trade-off [3]. The

trade-off is between the number of bits used to represent the image (the rate) and the fidelity of the reconstructed representation (the distortion).

Lossy encoding is defined as a data compression algorithm that loses some of the input data during encoding in order to achieve a better compression ratio. Some common lossy compression schemes are predictive coding and transform coding [32].

2.2.2 Video Coding Schemes

Digital video coding technology has diversified and it is targeted for a plethora of emerging applications, such as video on demand and digital TV/HDTV broadcasting. The rising commercial interest in video communications posed the need for international image and video coding standards. Today, there are various video coding standards available. These standards were developed in conjunction with the development in coding, transmission, video storage and VLSI design technologies.

The Study Group XV of the Community Colleges for Innovative Technology Transfer (CCITT)[1] proposed the first international standard for video coding, H.120, in 1984[8]. With advances in image compression techniques and implementation technology, other standards have been proposed and been more commercially successful. These include H.261, H.263, Motion JPEG, MPEG-1, MPEG-2 and MPEG-4 [33],[31]-[10],[16]. The JPEG and MPEG

---

[1] The CCITT was re-named as the International Telecommunications Union – Telecommunications (ITU-T) sector in 1993.

visual coding standards were developed under the auspices of the International Standard Organization (ISO). In section 2.3, we review some of the characteristics of the MPEG video coding standards.

2.2.3 Distortion Measures

Once an image has been converted into a digital format and is subjected to lossy compression, the question of how to measure the trade-off in the fidelity of the decompressed representation arises. Extensive and continuing studies have been performed for a suitable measure of the distortion for images and video. An objective distortion measure that is widely used and accepted is the mean-squared error (MSE). It has been shown that systems that are optimized for MSE performance also give fairly good results in terms of the perceptive quality of the image [12][40]. The Mean-Squared Error is the sample average of the total squared error between the compressed and the original image. It is mathematically represented as follows:

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}\left[I(x,y) - I'(x,y)\right]^2 \qquad (2.2)$$

where, $I(x,y)$ is the original image, $I'(x,y)$ is the decompressed image and M, N are the dimensions of the image.

Another metric that is used for evaluating the performance of an image compression technique is the *peak signal-to-noise ratio* (PSNR). The PSNR is closely related to the MSE and is expressed in terms of the MSE as:

$$PSNR = 20 * \log_{10}\left(255/\sqrt{MSE}\right) \quad (2.3)$$

A lower value of MSE implies a lesser error in the decompressed image. Inversely, a high PSNR is desirable because it implies that the ratio of signal to noise is higher where the 'signal' is the original image and the 'noise' is the error in reconstruction.

2.3 MPEG Video Coding Standard

The Moving Picture Experts Group (MPEG) was founded in 1988 to standardize video coding algorithms for digital storage media at bit rates up to about 1.5 Mbits/s [19]. However, beginning with MPEG-2, the standards have been more generic. This means that the standard is not application specific and mainly consists of a set of tools that can be tailored to meet the needs of the user. In this section, we review some important characteristics of the MPEG framework.

2.3.1 Compression

Video consists of sequences of still pictures or frames that, if uncompressed, have data volumes that are extremely high. In the MPEG standard, each frame is divided into an array of 16x16 pixels, called a *macroblock*. Each macroblock comprises of four 8x8 luminance (intensity) blocks (or Y-blocks) and two 8x8 chrominance (color) blocks (Figure 2-2). The chroma blocks are each one U-block (blue) and one V-block (red). The color information, therefore, has half the horizontal and vertical resolution as the luminance Y information (ITU-R 601). This component video standard is known as the 4:2:0 standard. A picture with

a luminance resolution of 360 lines and 288 pixels per line is known as the *common interchange format* (CIF).

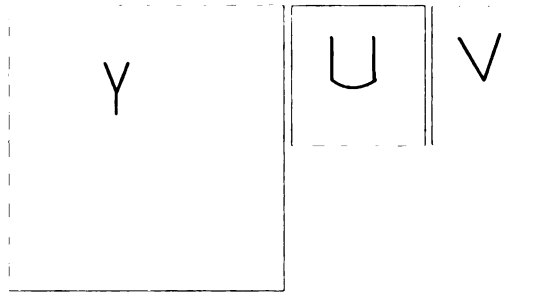


Figure 2-2 4:2:0 composite color standard

Popular video coding standards (including MPEG) employ a hybrid Motion-Compensation (MC) prediction and transform-domain DCT-based compression method. First, a macroblock at a given frame is predicted from one or two adjacent frames using Motion Estimation (ME). The prediction error is then transformed (using DCT), quantized, and entropy-coded. The type of prediction that is used for a particular frame defines different picture types as explained below.

2.3.2 Frame Types

MPEG video sequences consists of frames of three types:

Intra-coded Frames: (I Frames): The I frames use DCT encoding to compress one single frame, without being dependent on any other frame. I frames are inserted once every 12-15 frames and are used to start a sequence. Decoding of video can only start at an I frame.

Predicted Frames: (P Frames): P frames use motion compensation and DCT encoding to "predict" the values of each pixel. These frames are coded as the difference from the last I or P frame. They can offer a better compression ratio than I frames, depending on the amount of motion present.

Bi-directional Frames: (B Frames): B frames also use motion compensation and DCT encoding to estimate the value of each pixel. They differ from the P frame in the sense that they can use either the *last* I or P frame or the *next* I or P frame to get the value of each pixel.

Quantization further improves the compression ratio in all the frames. It is also useful to note, at this juncture, that since B Frames use both previous and subsequent frames for correct decoding, the temporal sequence of the frames in MPEG video differs from their spatial sequence. Figure 2-3 illustrates the types of frames and their sequential orders in space and time.

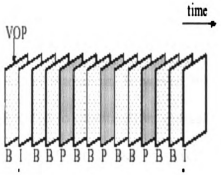


Figure 2-3 I, P and B Frames in MPEG [30]

2.3.3 Transform Coding

In MPEG video coding, the Discrete Cosine Transform (DCT) is used as part of the video compression algorithm.

The DCT is a mathematical transformation of a 2D matrix of pixel values into an equivalent matrix of spatial frequency components. It is an invertible, discrete, separable orthonormal transformation. The DCT can be mathematically represented as:

$$B(k_1,k_2)=\left(\frac{2}{N_1}\right)^{1/2}\left(\frac{2}{N_2}\right)^{1/2}\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}\Lambda(i).\Lambda(j)\cos\left[\frac{\pi k_1}{2N_1}(2i+1)\right]\cos\left[\frac{\pi k_2}{2N_2}(2j+1)\right].f(i,j)$$

where,

$$\Lambda(\xi)=\begin{cases}\frac{1}{2},\xi=0\\ 1,otherwise\end{cases}$$

(2.4)

and A is the input image, B is the output image, $A(i,j)$ is the intensity of the pixel in row i and column j within an $N_1 \times N_2$ image block, $B(k_1,k_2)$ is the DCT coefficient in row k_1 and column k_2 of the DCT matrix.

The DCT, by itself, is a lossless transform. However, when used as part of an image compression algorithm, it is subjected to quantization and subsequent run-length encoding. In MPEG, the DCT is used to transform blocks of pixel values into blocks of spatial frequency values. These blocks are organized in such a way that the lower-frequency

components lie in the upper-left corner and the higher-frequency blocks lie in the lower-right corner. By this, the high frequency components are first dropped or discarded. Once DCT has been applied to an image, no spatial localization can be recovered. Also, canceling out sets of frequencies can lead to high distortion in image regions where the low frequency components have low amplitude. In order to overcome these shortcomings, the compression algorithm in MPEG applies the DCT to 8x8 blocks. This speeds up the algorithm and also, localizes the frequency content.

2.4 MPEG-4 Scalability Structure

Scalable video coding solutions have been used to facilitate delivery of multimedia content over networks with varying channel characteristics and unpredictable quality-of-service measures (e.g. the Internet)[7]. All types of scalable video consist of a base layer and one or more enhancement layers. The base layer represents the minimum data needed for decoding the video, so as to obtain a minimum video quality. The enhancement layer represents additional information that, if sent to the decoder, enhances the video quality.

Scalable video solutions are usually based on some scalability structure. This structure defines the relationship between the base layer and the enhancement layer frames of the video sequence. One scalability structure supported by MPEG-4 is *fine-granular scalability*. This structure allows for progressive decoding. This means that the decoder can start decoding once it has the base layer and any or no part of the enhancement layer. As more data is received, the quality of the decoded image improves. This process continues until the entire enhancement layer is received.

The FGS scheme in MPEG-4 is based on the scalability structure proposed in [17]. FGS is essentially a hybrid video-scalability solution. It has a prediction-based, motion-compensated base layer and a fine granular scalable enhancement layer[17]. This is illustrated in Figure 2-4. These characteristics allow it to cater to the dual requirement of efficient coding and continuous fine-grained scalability.

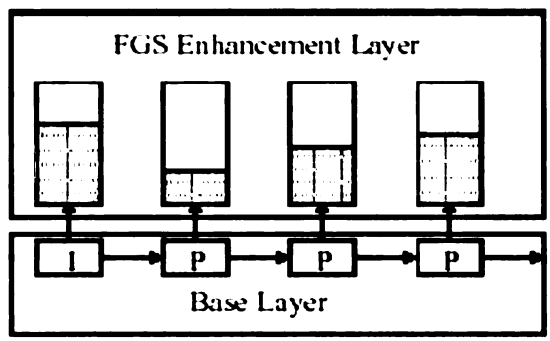


Figure 2-4 FGS scalability structure [30].

2.5 MPEG-4 Video Coding Framework

A typical system configured for streaming video over the Internet has the basic blocks of an encoder, streaming server, channel and decoder. This basic block diagram is shown in Figure 2-5.

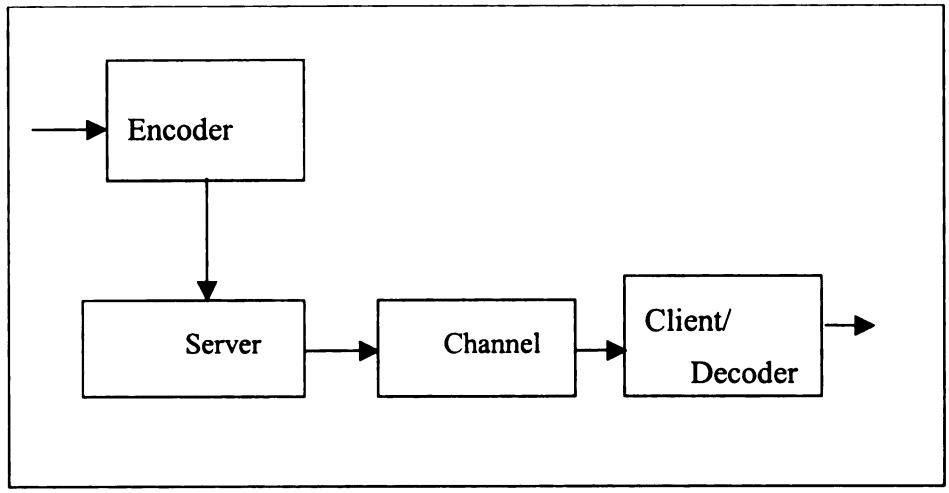


Figure 2-5 System configuration for streaming video over the Internet [42]

In this section, we shall look at the structure of these blocks in the MPEG-4 FGS framework. We proceed in logical order, starting from the encoder and ending with the decoder.

2.5.1 The Encoder

The detailed block diagram for the MPEG-4 FGS encoder is shown in Figure 2-6. The FGS framework requires the encoder to have two encoders- one to generate the base layer

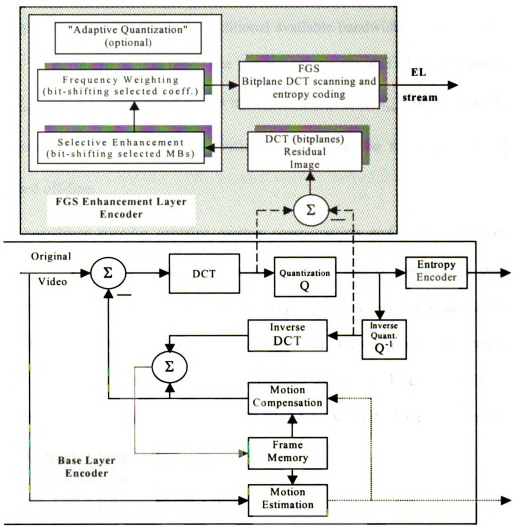


Figure 2-6 Basic (SNR) encoder for the FGS base and enhancement layers. It is clear that the added complexity of the FGS enhancement-layer encoder is relatively small [18].

bitstream and the other to generate the enhancement layer bitstream. The base layer is coded to a minimal acceptable video quality. In other words, the minimum available bandwidth must always be at least equal to the bitrate chosen for the base layer. This ensures that the reconstructed video has a quality equal to that provided by the base layer bitstream.

Thus, the base layer is coded to the minimum available bandwidth ($R_{BL} \sim R_{\min}$). The enhancement layer fully utilizes the additional available bandwidth at any instance of time. This can be estimated in real-time. The upper limit on the size of the enhancement layer ($R_{EL} = R_{\max} - R_{BL}$) is determined by the maximum bandwidth $R_{\max}$ that is available at time of transmission. An important observation here is that the range $[R_{\min}, R_{\max}]$ can be determined off-line.

The base-layer encoder contains motion compensation and motion estimation tools. Hence, the base layer bitstream is organized into I, P and B frames. The base layer is compressed using the DCT-based video encoding method. This method demonstrates good coding efficiency, particularly at low bitrates. The DCT coded video is quantized and sent to an entropy encoder to form the base layer bitstream. In the base layer, the DCT coefficients are run-length coded. The video coding tools used for coding the base layer are described in length in [38], [37], [20], [22].

Encoding The Enhancement Layer

The FGS enhancement layer takes two inputs- the original frame and the reconstructed base-layer frame. The difference in the intensity value of every pixel in the original frame and the corresponding pixel in the reconstructed frame is DCT coded. The DCT coding follows the traditional 8x8 blocks described above. Also, each set of four 8x8 luminance (Y) blocks and two color (U and V) blocks form the typical 16x16 macroblock structure.

Once all the DCT coefficients in a frame have been obtained, the one with the highest absolute value is determined. The number of bits required to represent this maximum value

gives the maximum number of bit *levels* or *bitplanes* required to code the enhancement-layer (residual) frame. The absolute values of the coefficients in every 8x8 DCT block are zigzag ordered into an array. Consider an array of 64 bits representing each of these coefficients at a particular significant position say p. This array is said to be a *block bitplane* at level p.

Every frame has all the three-color components (Y, U and V) coded into it. The maximum number of bitplane levels required to code each of these components may differ. Mathematically [18],

$$N_{BP} = \lfloor \log_2 (|C|_{\max}) \rfloor + 1 \tag{2.5}$$

where, N_{BP} is the number of bitplanes and $|C|_{\max}$ is the maximum DCT magnitude of the given residual frame.

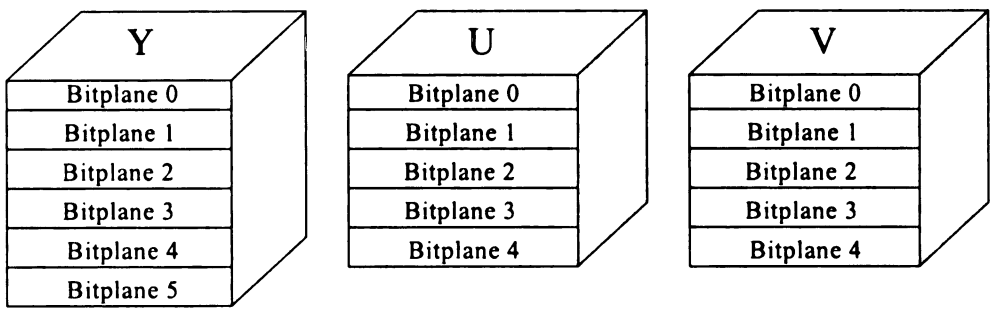


Figure 2-7 An example of different number of bitplane levels for the different color components.

The maximum number of bitplanes required to code the frame under consideration is that corresponding to the greatest of the maximum levels of Y, U and V. Hence, not all the blocks have the same number of bitplane levels. This is illustrated in Figure 2-7.

The maximum number of bitplanes required to code the frame under consideration is that corresponding to the greatest of the maximum levels of Y, U and V. Hence, not all the blocks have the same number of bitplane levels. This is illustrated in Figure 2-7.

Each such block is now subject to variable-length coding (VLC). The code length depends on the significant position of the bitplane level under consideration. It must be noted here that this is the *absolute* significant position of the level in the binary representation of the largest DCT coefficient in the given block.

The number of passes required to code a particular block depends on the number and order of 1s and 0s in each block.Figure 2-8 illustrates an example that helps understand this better.

0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0

Figure 2-8 an example of the number of passes required to code a block. This number depends on the arrangement of 1s and 0s in the block. A continuous, uninterrupted string of zeros is called a *run*. Here, number of runs =4.

More information on the bitplane coding method used for the FGS enhancement layer can be found in [14],[18],[42],[21].

2.5.2 The Streaming Server

The second block in the Internet streaming video system structure is the streaming server. In the case of the MPEG-4 FGS system, the streaming server determines the size of the portion of the enhancement layer that gets transmitted. Figure 2-9 shows the total, lossless enhancement layer being input to the server. The shaded region indicates the part of the enhancement layer of every frame that fits within the available bandwidth. As shown in the figure, only this part of the enhancement layer reaches the decoder.

In the current implementation of the MPEG-4 FGS standard, the streaming server allocates bitrates for the enhancement layer of every frame by a brute-force method. The basic equation defining this allocation is:

$$B_{FR}(EL) = {}^{B_{TOT}(EL)}\!/_{N_{FR}} \tag{2.6}$$

where, $B_{FR}(EL)$ is the bitrate allocated for a given frame in the enhancement layer, $B_{TOT}(EL)$ is the total bitrate available for the enhancement layer of the video sequence and N_{FR} is the number of frames in the video sequence.

Within a frame, the streaming server performs a crude, raster-scan driven pruning of the enhancement layer to fit the bitrate allocated for that frame. This is a low-complexity, non-optimal bit allocation method. The objective of this thesis is to develop and analyze the performance of optimum (in a rate-distortion sense) algorithms for FGS rate control.

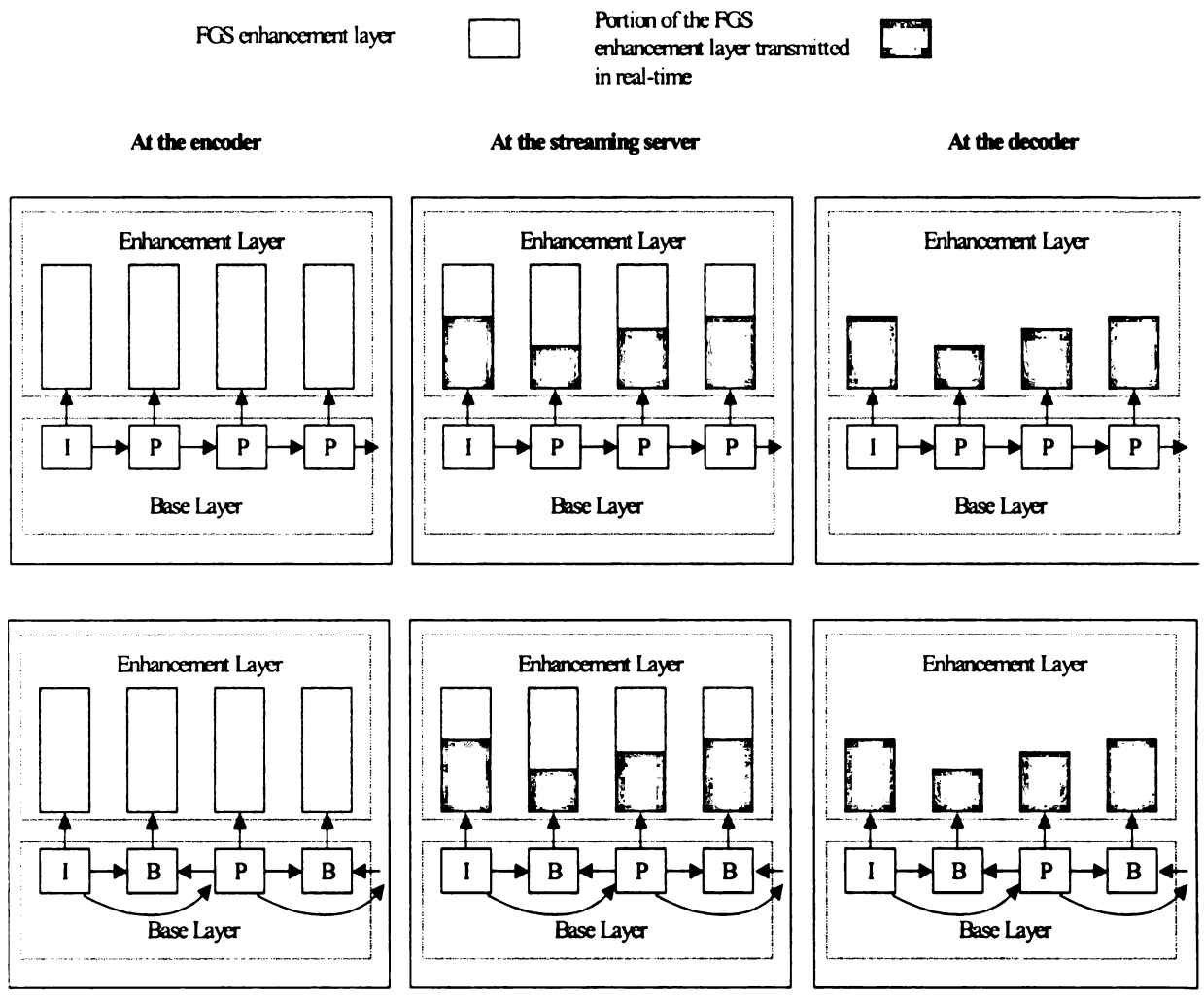


Figure 2-9 Examples of the FGS scalability structure at the encoder (left), streaming server (center), and decoder (right) for a typical unicast Internet streaming application. The top and bottom rows of the figure represent base-layers without and with Bi-directional (B) frames, respectively. Note that only part of the enhancement layer is transmitted in real time. [18]

2.5.3 The Decoder

The last block in the Streaming server system is the decoder. It is the client-side module. The block diagram for an MPEG-4 FGS decoder is shown in Figure 2-10 below. The

structure of the decoder is basically the reverse of that of the encoder. The decoder reconstructs the video frames using the base layer bitstream and the truncated enhancement layer bitstream. It is useful to note that the two bitstreams are decoded independently and added to form the reconstructed frame. The quality of the video is proportional to the number of bits decoded by the decoder for the enhancement layer of each frame.

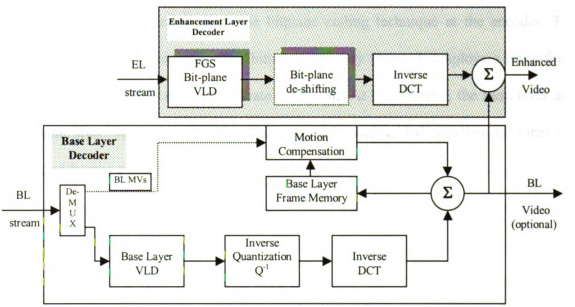


Figure 2-10 Basic (SNR) FGS decoder for the base and enhancement layers [18].

2.6 Summary

To summarize, rate-distortion theory indicates that there exists a minimum distortion level that can be associated with any given bitrate lesser than the total lossless bitrate. In the

case of image data compression, the losses incurred due to applying an image compression algorithm lead to a similar trade off between rate and distortion. A widely used measure of distortion for image data is the Mean Squared Error.

The compression algorithm in the MPEG-4 video-coding standard comprises of transform coding using the DCT, quantization and run-length encoding. The FGS scalability scheme allows for progressive scalability in the enhancement layer. This is possible because the enhancement layer is coded using the bitplane coding technique at the encoder. The streaming server roughly determines the bitrates for every frame. It employs a brute-force approach to allocate bits to meet this bitrate constraint. At the decoder, the base layer and enhancement layer bitstreams are decoded separately and added together to form the reconstructed frames of the video sequence.

3 Bitplane Rate-Distortion Based Rate-Control Algorithm

In this chapter, we describe and develop a rate-distortion based rate-control algorithm for the *specific* encoder-decoder system of the MPEG-4 FGS standard. This makes it an *operational* rate-distortion based algorithm. The advantages of using an operational curve is that the data points used are directly achievable for the implementation of the system and for the given test data.

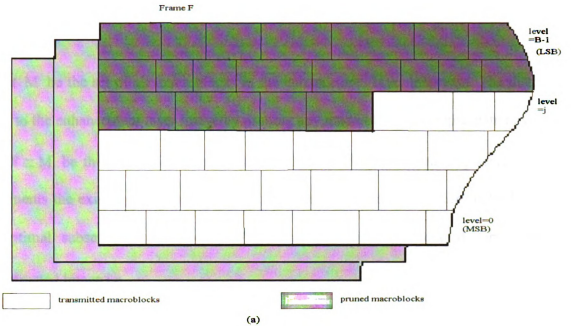
FGS codes a video sequence into a base layer and a single enhancement layer. The base layer is non-scalable and is equal in size to the lower bound of the available bandwidth. The residual DCT coefficients representing the difference between the original picture and the reconstructed picture are bitplane coded and comprise the enhancement layer. The enhancement layer can be truncated within each frame to allow partial enhancement proportional to the number of bits decoded for every frame.

The limiting of the base layer bitrate to the minimum available bandwidth is a typical rate-control problem. Rate control for non-scalable coders is essentially based on a buffering technique. The quantization step size is based on the fullness of the output buffer and the characteristics of the picture to be coded [44], [4], [13]. MPEG-4 uses the rate control scheme proposed for MPEG-2 and which is commonly known as the Test Model 5 (TM5). This rate control algorithm consists of three steps- target bit allocation for a frame, rate control via buffer monitoring and adaptive quantization based on local activity [39]. The

TM5 algorithm is used to restrict the base layer to the lower bound of the available bandwidth.

The enhancement layer bitrate depends on the bandwidth that is available for the video sequence streaming above the base layer bitrate. The encoded enhancement layer bitstream is sent to the streaming server. Effectively, the streaming server performs a raster scan of the bitplanes- starting from the first macroblock through to the last on a particular bitplane. The first bitplane scanned is the one that corresponds to the most significant position of the largest DCT value in that frame. Once all the macroblocks in a particular bitplane have been sent, the server starts sending the macroblocks from the beginning of the next highest significant position bitplane. This is illustrated in Figure 3-1. The server stops sending more enhancement layer information whenever the target available bitrate is reached for that frame. This is a coarse raster-scan bit allocation method that has very low complexity. However, such a method is not optimal since it performs brute-force bit-allocation.

When the streaming server prunes macroblocks from the enhancement layer bitstream, it adds to the distortion in the reconstructed enhancement layer image. The rest of this chapter is devoted to investigating a low computation, low complexity rate-control algorithm for the FGS enhancement layer that minimizes this distortion. It is also intended that this algorithm be compatible with the rate-control mechanism employed for the base layer.



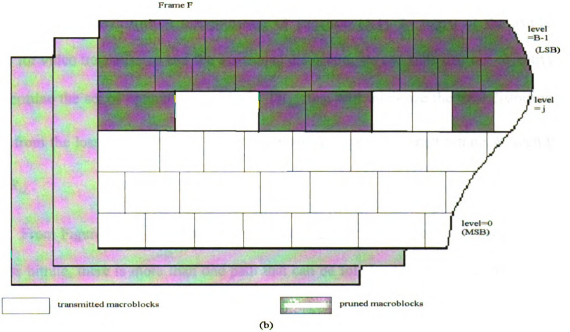


Figure 3-1 Pruning of macroblocks- width of the macroblocks represents the number of bits needed to encode that macroblock (a) Raster-scan rate-control (b) R-D based rate-control

3.1 Rate And Distortion Measures

Let M be the set of all macroblocks in the enhancement layer bitstream. The distortion added to the enhancement layer image by pruning any macroblock m is represented by ΔD_m. Let $\mathfrak{M} \subset \mathrm{M}$ be the subset of macroblocks that is pruned by the streaming server when it implements the existing raster-scan rate-control method. Our objective here is to determine the optimal subset of macroblocks, $\mathfrak{M}_{opt}$, that minimizes the total distortion in the reconstructed image i.e.,

$$\mathfrak{M}_{opt} = \arg\left\{\min_{\mathfrak{M}} \sum_{m \in \mathfrak{M}} \Delta D_m\right\} \tag{3.1}$$

We make a rate-distortion interpretation of equation (3.1). In terms of the available bitrate for video transmission and the associated distortion, we can state that our objective is to determine the set of optimal macroblocks $\mathfrak{M}_{opt}$ that will minimize the total distortion in going from the lossless bitrate R_{LL} of the enhancement layer to a target bitrate R_T such that $R_T \leq R_{LL}$.

From Figure 3-2, we can see that in order to get to any bitrate that is lesser than the lossless bitrate, there is more than one path that can be followed. To find the path with the least associated distortion at all points or the optimal performance curve, we have to find the one that will have minimum slope at every point. Hence, in a rate-distortion sense, our objective is to minimize the ratio between the increase in the distortion associated with a given reduction in the rate. In terms of the macroblocks to be pruned, our objective is to

determine the set of macroblocks that, when pruned, result in the minimum slope of the rate-distortion curve. This implies that we have to determine the rate-distortion ratio associated with the pruning of each macroblock in the frame.

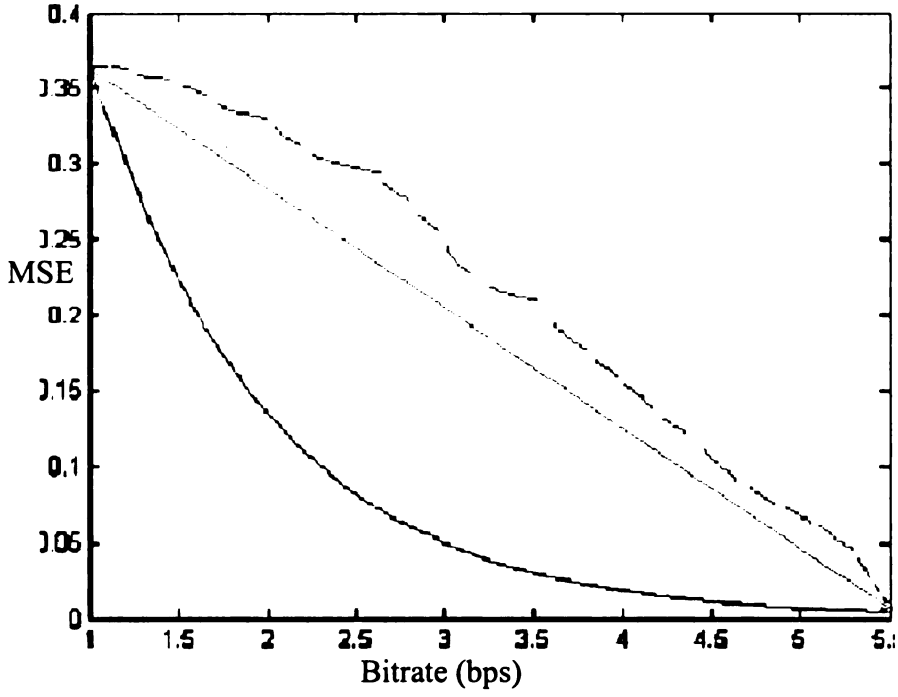


Figure 3-2 Different rate-distortion curves

In order to find the subset $\mathfrak{M}_{opt}$, we have to find the distortion ΔD_m associated with the pruning of each macroblock in the frame. One method of doing this is to completely decode the enhancement layer bitstream to get the reconstructed image. This image is then compared with the original image and the mean-squared error (MSE) between the two is determined. This gives us the distortion in the pixel domain. In the CIF format, the size of a frame is 352 x 288 pixels. Each frame can hence be divided into 396 macroblocks (16x16

pixels). If B is the number of bitplanes in the enhancement layer frame, then it contains a total of $396*B$ macroblocks. Calculating the distortion associated with each of these macroblocks by reconstructing the image completely, results in very high computational complexity. This is neither desirable nor acceptable. Hence, we look for an alternative method of determining ΔD_m.

One possible alternative is to determine the MSE in the transform domain. Let $\left[X(\bar{k})\right]$ be the transform domain representation of the original image $\left[x(\bar{n})\right]$ and $\left[\hat{X}(\bar{k})\right]$ be the transform domain representation of the reconstructed image $\left[\hat{x}(\bar{n})\right]$. Here, $\bar{k} = (k_1, k_2)$ and $\bar{n} = (n_1, n_2)$ are two-dimensional vectors in the transform and pixel domains, respectively. It can be easily shown that the squared error in the transform domain is equal to the squared error in the pixel domain, that is, the distortion,

$$D_{l,m} = \sum_{\bar{k}} \left[X_{l,m}(\bar{k}) - \hat{X}_{l,m}(\bar{k}) \right]^2 = \sum_{\bar{n}} \left[x_{l,m}(\bar{n}) - \hat{x}_{l,m}(\bar{n}) \right]^2 \qquad (3.2)$$

where, $D_{l,m}$ is the distortion associated with block l of macroblock m, $X_{l,m}(\bar{k})$ and $\hat{X}_{l,m}(\bar{k})$ are the DCT coefficients in position $\bar{k} = (k_1, k_2)$ in the same block in the transform domain representations of the original and reconstructed images respectively. $x_{l,m}(\bar{n})$ and $\hat{x}_{l,m}(\bar{n})$ are the corresponding intensity values of the pixel at location $\bar{n} = (n_1, n_2)$ in block l of macroblock m. Therefore; we use the DCT coefficients in the enhancement layer to determine the distortion.

If l is the number of blocks in the macroblock m, the distortion for the entire macroblock is given by,

$$D_m = \sum_l D_{l,m} \tag{3.3}$$

Our objective is to determine the set of optimal macroblocks $\mathfrak{M}_{opt}$ that will minimize the total distortion in going from the lossless bitrate R_{LL} of the enhancement layer to a target bitrate R_T such that $R_T \leq R_{LL}$. The macroblocks that we select for pruning have to satisfy the condition,

$$\sum_{m \in \mathfrak{M}} R_m \geq (R_{LL} - R_T) \tag{3.4}$$

Therefore, the optimization RD-based problem addressed in this thesis can be expressed as follows:

$$\mathfrak{M}_{opt} = \arg\left\{ \min_{\substack{\mathfrak{M} \\ \sum_{m \in \mathfrak{M}} R_m \geq (R_{LL} - R_T)}} \left(\sum_{m \in \mathfrak{M}} \Delta D_m \right) \right\} \tag{3.5}$$

3.2 The Rate Control Algorithm

In this section, we introduce the bitplane rate-distortion based rate-control (BPRD) algorithm for the FGS enhancement layer.

3.2.1 Mathematical Formulation

Let the target bitrate for the FGS enhancement layer be R_T. Let $R_i, i = 0,1,...B-1$ be the number of bits required to code the ith bitplane. Here i=0 corresponds to the MSB bitplane and i=(B-1) corresponds to the plane representing the least-significant bit (LSB). In terms of R_i, the target bitrate can be expressed as,

$$\sum_{i=0}^{j-1} R_i \le R_T \le \sum_{i=0}^{j} R_i \tag{3.6}$$

where the target bitrate R_T lies between the number of bits required to encode j bitplanes of the image and the number of bits required to encode j-1 bitplanes of the image. The existing raster-scan rate-control method completely discards the (j+1)st to the (B-1)st bitplanes. Macroblocks are pruned from the bitplane level j in order to reach the target bitrate R_T. In the bitplane rate-distortion based rate-control (BPRD) algorithm described in this section, we therefore resolve to find the optimal set of macroblocks that must be pruned from bitplane level j. This is given by

$$\mathfrak{M}_{opt(j)} \subset \mathrm{M}_j \tag{3.7}$$

where the global set of macroblocks is considered to be M_j, the set of all macroblocks at bitplane level j.

Given a target bitrate R_T, the rate-control algorithm begins at a bitplane level j. The input to the rate-control algorithm is hence considered to be the video sequence at the bitrate

$\sum_{i=j}^{B-1} R_i$. In other words, the input video that the distortion is measured against is considered to have only the j most-significant bitplane levels if the algorithm is operating on the bitplane level j.

In the FGS enhancement layer, the DCT coefficients are bitplane coded. In these terms, the DCT coefficient in block l of macroblock m can be represented as,

$$X_{l,m}(\bar{k}) = \sum_{i=0}^{B-1} 2^{(B-1)-i} b_{l,m,i}(\bar{k}) \tag{3.8}$$

where, $b_{l,m,i}(\bar{k})$ is the bit in the ith significant position of the coefficient $X_{l,m}(\bar{k})$ and B is the total number of bitplane levels needed to represent the block l of macroblock m.

In the case of the proposed BPRD, there is always only one bitplane in the binary representation of the DCT coefficient, that is, $B = 1$ always. This means that the index i=0 always. For this reason, we shall henceforth drop the index i. Equation (3.8) thus reduces to,

$$X_{l,m}(\bar{k}) = b_{l,m}(\bar{k}) \tag{3.9}$$

The distortion $D_{l,m}$ (mean-squared error) between the block l in the macroblock m in the transform domains of the original and the reconstructed images is given by,

$$\sum_{\bar{k}} \left[X_{l,m}(\bar{k}) - \hat{X}_{l,m}(\bar{k}) \right]^2 = \sum_{\bar{k}} \left[b_{l,m}(\bar{k}) - \hat{b}_{l,m}(\bar{k}) \right]^2 \tag{3.10}$$

Since there is a distortion already associated with the input image, this distortion can be considered to be the increase from the original distortion of the input image or $\Delta D_{l,m}$. If $j = B$, then the input distortion is 0.

Using equation (3.3), the total change in distortion due to the pruning of the macroblock m (ΔD_m) is given by,

$$\Delta D_m = \sum_l \sum_{\bar{k}} \left[X_{l,m}(\bar{k}) - \hat{X}_{l,m}(\bar{k}) \right]^2 = \sum_l \sum_{\bar{k}} \left[b_{l,m}(\bar{k}) - \hat{b}_{l,m}(\bar{k}) \right]^2 \qquad (3.11)$$

Here, $b_{l,m}(\bar{k})$ and $\hat{b}_{l,m}(\bar{k})$ can take values of 0 or 1. This implies that $b_{l,m}(\bar{k}) - \hat{b}_{l,m}(\bar{k})$ is also always 0 or 1. Hence, we can rewrite equation (3.11) as,

$$\Delta D_m = \sum_l \sum_{\bar{k}} \left[X_{l,m}(\bar{k}) - \hat{X}_{l,m}(\bar{k}) \right]^2 = \sum_l \sum_{\bar{k}} \left[b_{l,m}(\bar{k}) \right]^2 = \sum_l \sum_{\bar{k}} b_{l,m}(\bar{k}) \qquad (3.12)$$

Only the cases in which $b_{l,m}(\bar{k}) = 1$ and $\hat{b}_{l,m}(\bar{k}) = 0$ contribute to this sum. Hence, the distortion added by pruning a macroblock m at bitplane level j is equal to the number of ones in the macroblock at level j that get converted to zeros. This is the measure of distortion for the BPRD algorithm.

Thus, a simple indication of the distortion is the number of ones in a bitplane macroblock. Every block in a macroblock comprises of ones and zeros. Each such block is separately VLC coded based on the number of consecutive zeros (run of zeros) and the length of these runs. The number of bits required to encode a particular block depends on the bitplane level and the length and number of runs of zeros. From the VLC tables developed for the FGS enhancement layer in [21], the number of bits required to code a block that consists entirely of zeros (or an all-zero block) is less than that required to code a block that contains one or more runs. Hence, converting a block to the all-zero state reduces the bitrate. This difference in the bitrate is the change in the rate $\Delta R_{l,m}$ due to converting (to zeros) the

bitplane of block l of macroblock m. Therefore, the total number of bits reduced by pruning macroblock m:

$$\Delta R_m = \sum_l \Delta R_{l,m} \tag{3.13}$$

3.2.2 Description

The BPRD algorithm is *bitplane-level independent*. It is also frame specific. It operates only on the bitplanes in the current frame and not across frames. The steps involved are enumerated below:

Step 1: Start algorithm for frame f, bitplane level i such that $0 \le i \le B-1$;

Step 2: Recover the DCT coefficients from the compressed enhancement layer bitstream by performing variable-length decoding.

Step 3: Determine the change in rate (ΔR_m) and change in distortion (ΔD_m) for every macroblock m as described the previous section. Calculate the rate-distortion ratio $\left(\Delta D_m / \Delta R_m\right)$ for each macroblock.

Step 4: Store references to all the macroblocks in an array and sort them in ascending order of the rate-distortion ratio.

Step 5: Convert the macroblock referenced at the bottom of the sorted array to all-zero. Move reference one step at a time up the sorted order.

Step 6: Check if the target bitrate has been reached by this conversion.

Step 7: If the target bitrate has not been reached, go to *Step 4*.

Step 8: If the target bitrate has been reached, re-encode the modified DCT coefficients using variable-length coding.

Step 9: Repeat for all frames *f*.

Before proceeding to the experimental results, we highlight two important aspects of the proposed BPRD algorithm

The All-Zero Code: When the enhancement layer is VLC coded based on the MPEG-4 FGS standard; the code for every *block* depends on the significant position of the bitplane level under consideration. This level is the absolute significant position of the bitplane level in the binary representation of the largest DCT coefficient in the given block. The most-significant-bit (MSB) plane for every individual block is defined as the first non-all-zero bitplane. If an all-zero block is encountered after the MSB level has been reached for that block, it is coded using a special symbol called the *all-zero* symbol. The length of the all-zero symbol once again depends on the absolute significant position of its bitplane level, for that block.

More importantly, the length for the all-zero symbol used by the MPEG-4 FGS standard could be very high, and consequently not very efficient to use in our proposed BPRD algorithm. Consequently, we introduce into the enhancement layer bitstream, a flag to mark every macroblock in the bitplane under consideration. This flag is set to 1 if the macroblock has been chosen from the sorted order for conversion to all-zero. If not, the flag is set to

zero. The change in rate measure (ΔR_m) that is used above in our algorithm takes into consideration this extra (though very minor) overhead.

Blocks versus Macroblocks: Not every 8x8 DCT block has the same number of bitplanes. Each of the blocks is an all-zero case at bitplane levels higher than its MSB plane. At the encoder, if a block is all zero above its MSB level, a 1-bit flag is used to indicate that the MSB for that block has not been reached. It is possible that there are a lot of 8x8 DCT blocks that have fewer bitplanes than the maximum number of bitplanes in a frame [14]. In other words, the first couple of bitplanes of every frame are very likely to have many all-zero cases. In such a scenario, using 1 bit to signal the all-zero status of every block is not very efficient. The encoder instead groups the blocks in each macroblock together and uses a single bit to indicate an all-zero macroblock [14], [21].

Let us assume that our granularity resolution was a block instead of a macroblock. In the raster-scan bit allocation method, the blocks that lie beyond the target bitrate would be discarded. In other words, no overhead is added to the enhancement layer bitstream to indicate to the decoder that these blocks are to be rebuilt as all-zero cases. In the case of bitplane rate-distortion rate-control (BPRD), we would start from the bottom of the sorted array containing references to the blocks in the ascending order of their rate-distortion ratios and convert the blocks to all-zeros. We are not concerned with the position of the block within the frame. Our primary interest is to minimize the slope of the rate-distortion curve. However, it now becomes imperative that we signal the decoder that the block has been converted to an all-zero block and should hence be rebuilt as one. The relative overhead incurred in indicating this conversion on a block-by-block basis is high for every bitplane

level, but especially so for the first two-bitplane levels. This is because in these bitplane levels, the encoder itself has been optimized so that it adds the minimum number of bits into the enhancement layer bitstream. In order to restrict the overhead closer to the lowest possible level, we select a macroblock as the granular resolution of our rate-control algorithm.

3.3 Experimental Analysis

We begin the experimental analysis of the bitplane rate-distortion based rate-control algorithm by defining a suitable comprehensive test-bed.

Video Format

The file format used here is the common-interchange format (CIF). The frame size is therefore, 352 x 288 and the input source frame rate is 30Hz.

Test Sequences

The test sequences library is described in the MPEG-4 standard [21]. The library is divided into six classes from class A through to class F. The sequences are classified based on characteristics such as spatial detail, amount of movement, number of bits used for coding. The test sequences used in this experimental analysis belong to Class A through to Class C. Class A consists of sequences that have *low spatial detail* and *low amount of movement* (e.g. Akiyo). Class B sequences are characterized by *medium spatial detail* and *low amount of movement* or *vice versa* (e.g. Foreman). Class C sequences have *high spatial*

detail and *medium amount of movement* or *vice versa* (e.g. Mobile Calendar). The example sequences are the ones that are used here.

Base Layer Rate Control

We consider the following two cases for rate control in the base layer. First, we assume that there is no rate control algorithm applied to the base layer. We fix the quantization parameter Q_p and run the BPRD algorithm on the enhancement layer bitstream. The values we use for the test sequences are Q_p = 4, 6, 8 for the Akiyo sequence and Q_p = 10, 15, 28 for the Foreman and Mobile sequences. These Q_p values are chosen because they approximate to reasonable base layer sizes of 100 kbits, 200 kbits and 300 kbits respectively. Second, we consider the sequences with TM-5 rate-control applied to the base layer. Again, we restrict the base layer to the sizes of 100 kbits, 200 kbits and 300 kbits.

We consider the following two cases for rate control in the base layer. First, we assume that there is no rate control algorithm applied to the base layer. We fix the quantization parameter Q_p and run the BPRD algorithm on the enhancement layer bitstream. The values we use for the test sequences are Q_p = 4, 6, 8 for the Akiyo sequence and Q_p = 10, 15, 28 for the Foreman and Mobile sequences. These Q_p values are chosen because they approximate to reasonable base layer sizes of 100 kbits, 200 kbits and 300 kbits respectively. Second, we consider the sequences with TM-5 rate-control applied to the base layer. Again, we restrict the base layer to the sizes of 100 kbits, 200 kbits and 300 kbits.

We present below the results obtained by running the BPRD algorithm on the enhancement layer sequences obtained for all the different cases derived from the test-bed described above. All figures represent the mean-squared error plotted against the bitrate (in bits per second). The bitplane level=0 corresponds to the MSB bitplane.

In this section, four performance curves are presented for every case. The curves in the top-left corner represent the performance at all the bitplanes for both the current raster-scan and the proposed BPRD-based algorithms. As explained above, these results are shown for a given frame with the BPRD algorithm applied independently to each bitplane. The results are organized sequence-wise.

The first sequence that we analyze is the Akiyo sequence. This is a Class A sequence i.e. it has low spatial detail and low amount of motion. Figure 3-3-Figure 3-5 show the results obtained for the I, P and B frames for Q_p= 8 respectively. For the Akiyo sequence, a Q_p value of 8 corresponds to a base layer bitrate of about 100 kbits/s. Figure 3-6-Figure 3-8 show the result of applying the BPRD algorithm on the individual bitplanes of the I, P and B frames for the Akiyo sequence. The base layer in these cases is coded at 100 kbits/s using the TM-5 rate-control algorithm.

It can be seen from the top-left curve in Figure 3-3 that the performance curve of the existing raster-scan rate-control method closely follows that of the BPRD algorithm for bitrates higher than about 200 kbits/s per frame.

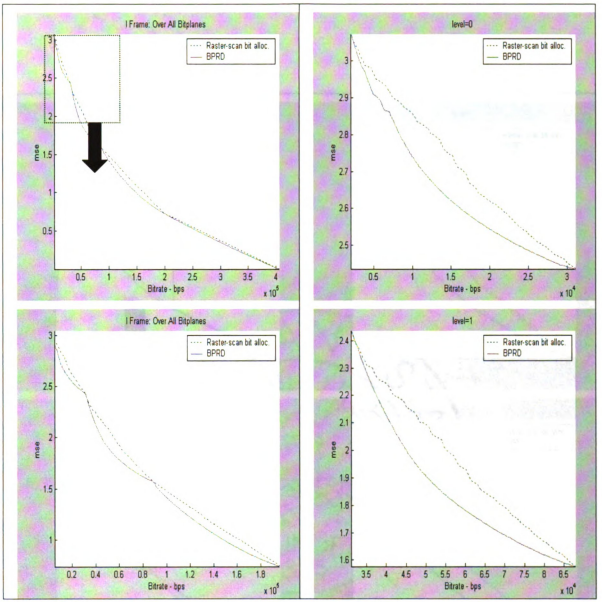


Figure 3-3 Akiyo I frame Qp=8

The improvement in performance is more significant for bitrates in the range of 50 kbits/s-200 kbits/s for each of the frames. At 10Hz, this corresponds to a total enhancement layer bitrate in the range of 500 kbits/s to 2 Mbits/s. This is a desirable and practical bandwidth range. The rest of the figure shows the performance in this specific bitrate range.

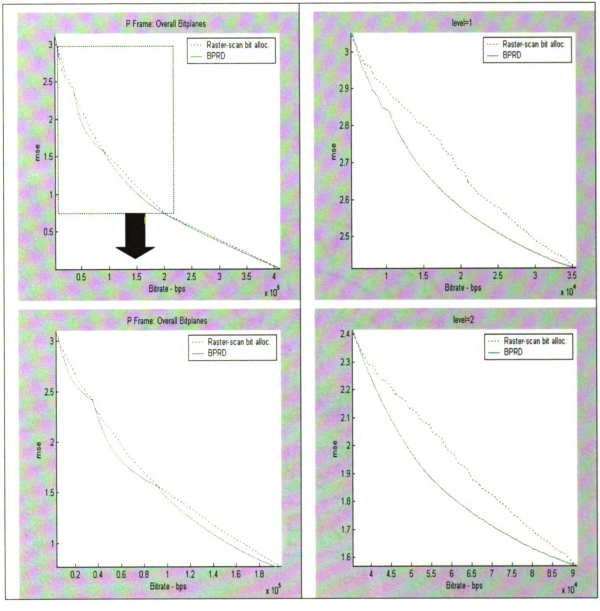


Figure 3-4 Akiyo P frame Qp=8

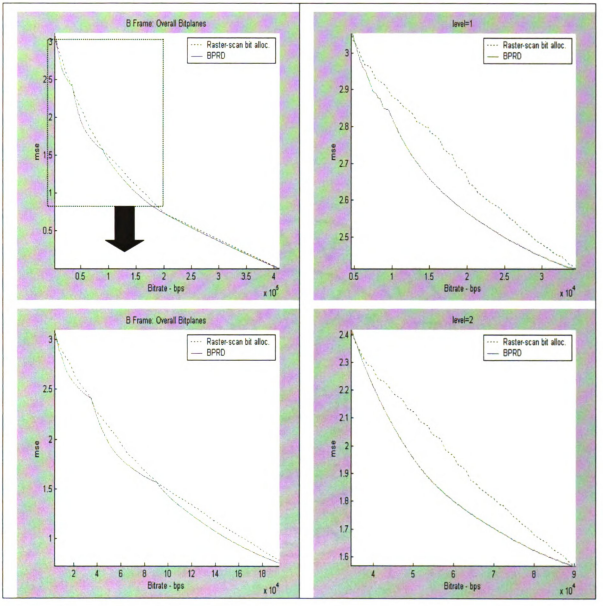


Figure 3-5 Akiyo B frame Qp=8

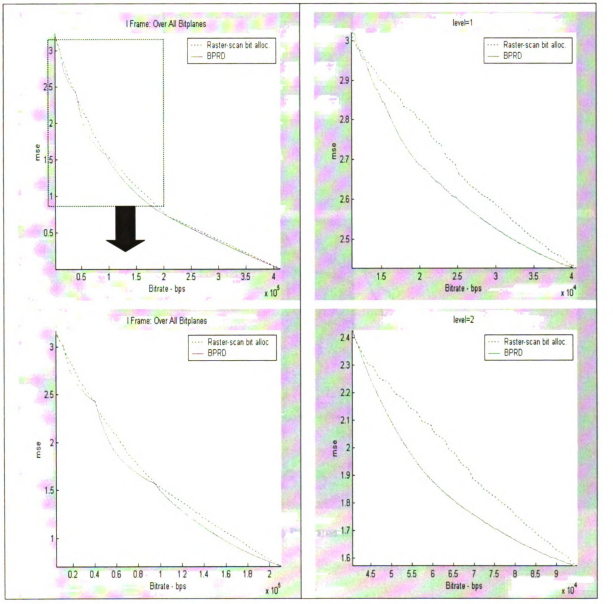


Figure 3-6 Akiyo I Frame Base layer=100k

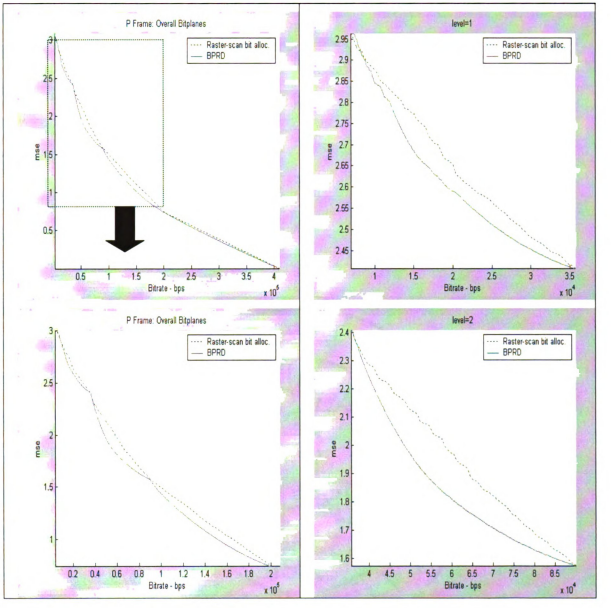


Figure 3-7 Akiyo P Frame Base layer=100k

Based on the above figure, the BPRD based algorithm provides its highest gain in performance at bandwidth ranges that can be available over current and evolving access networks (e.g., Internet access over Local-Area-Networks, cable modems, DSL, etc.).

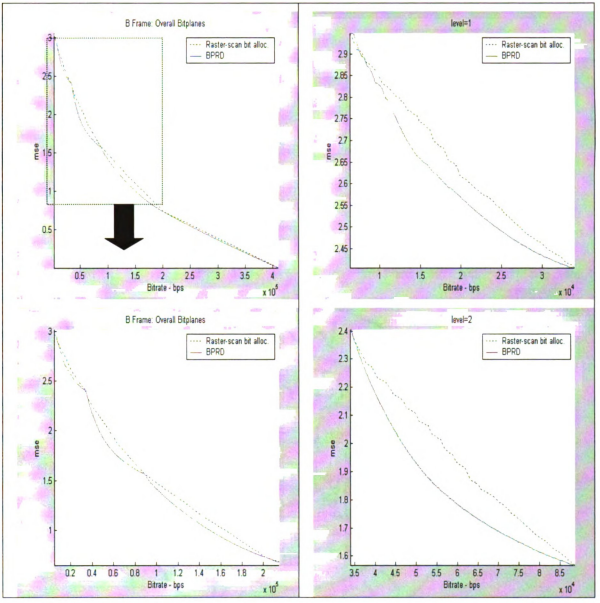


Figure 3-8 Akiyo B Frame Base layer=100k

From the figures shown in the following pages, it can be seen that this observation is applicable for all the presented cases for the Akiyo sequence. Hence, all the figures show specific results for the bandwidth range of 500 kbits/s to 2 Mbits/s. In all cases, this bitrate range corresponds to bitplane levels 1 and 2 for the Akiyo sequence.

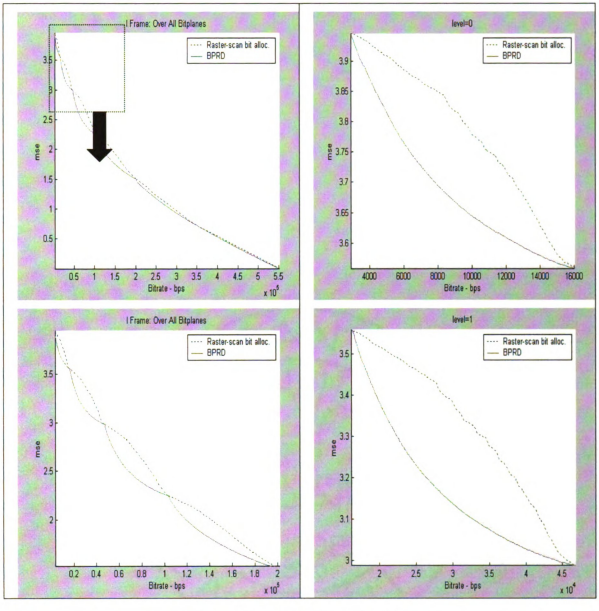


Figure 3-9 Foreman I frame for Qp=28

The second test-sequence used in this experimental analysis of the BPRD is the Foreman sequence. The Foreman is a Class B sequence. It has medium spatial detail and low amount of motion.

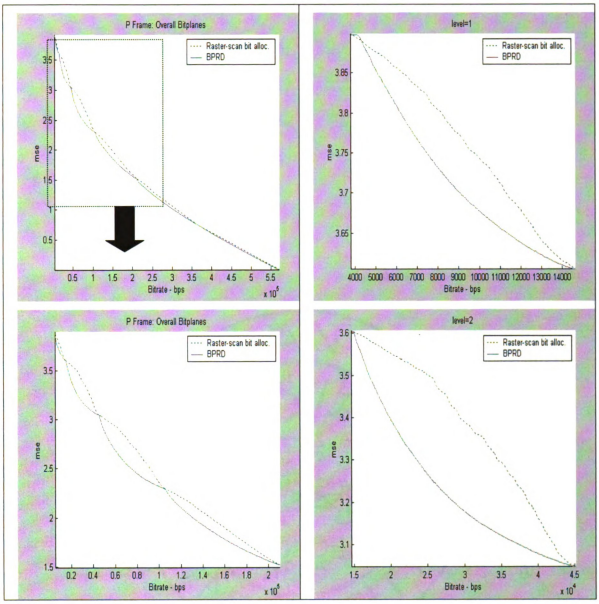


Figure 3-10 Foreman P frame for Qp=28

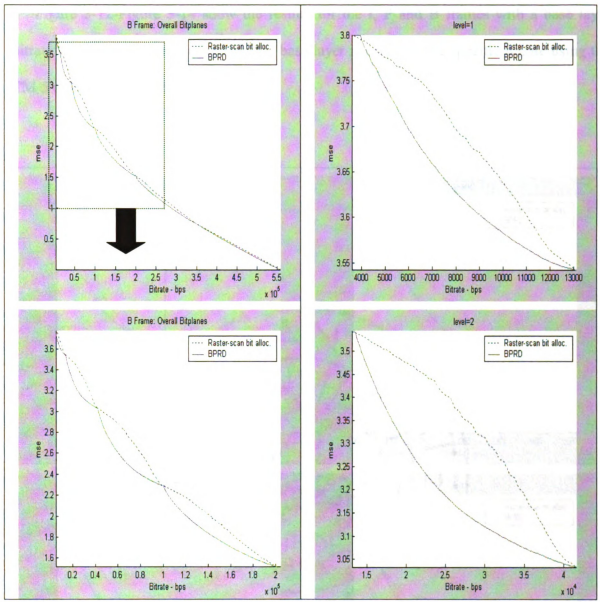


Figure 3-11 Foreman B frame for Qp=28

Figure 3-9 to Figure 3-11 show the results obtained by applying the BPRD algorithm to the individual bitplane levels coded at a constant Q_p value of 28 for the I, P and B frames respectively.

Figure 3-12-Figure 3-14 show the results for the I, P and B frames with a base layer bitrate of 100 kbits/s. In these cases, the base layer is coded to the specific bitrate using the TM-5 rate-control algorithm.

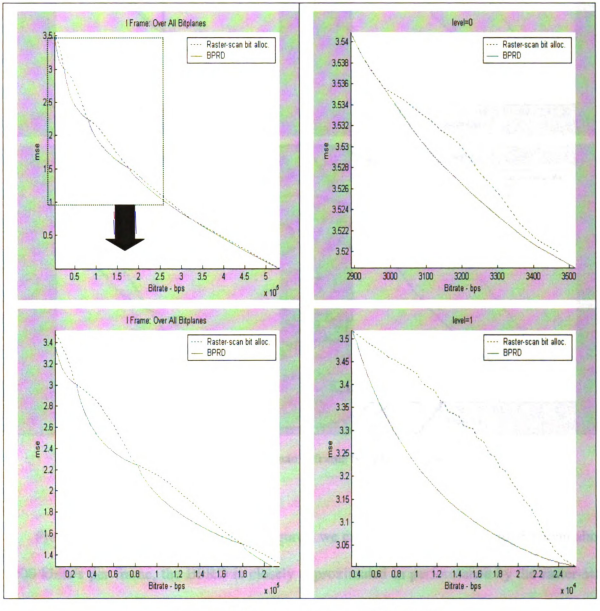


Figure 3-12 Foreman I frame Base layer=100k

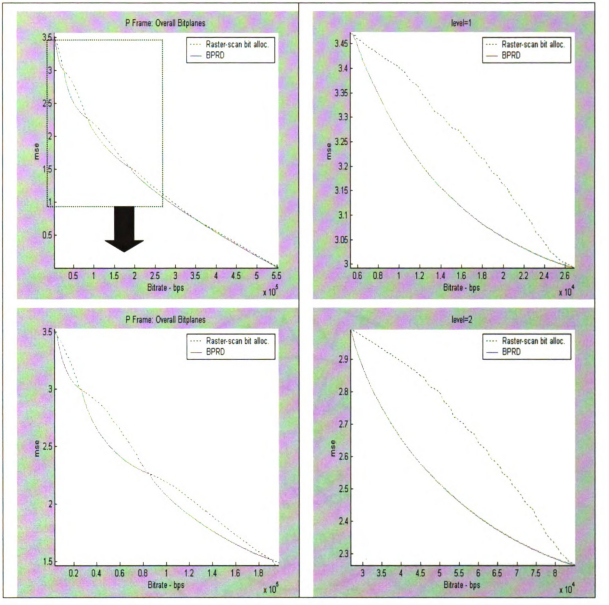


Figure 3-13 Foreman P frame Base layer=100k

From the top-left curve in all these figures, we can see that for bitrates higher than about 200 kbits/s per frame, the BPRD is closely approximated in performance by the raster-scan rate-control method. That is, for a 10Hz frame rate, the improvement in performance is more

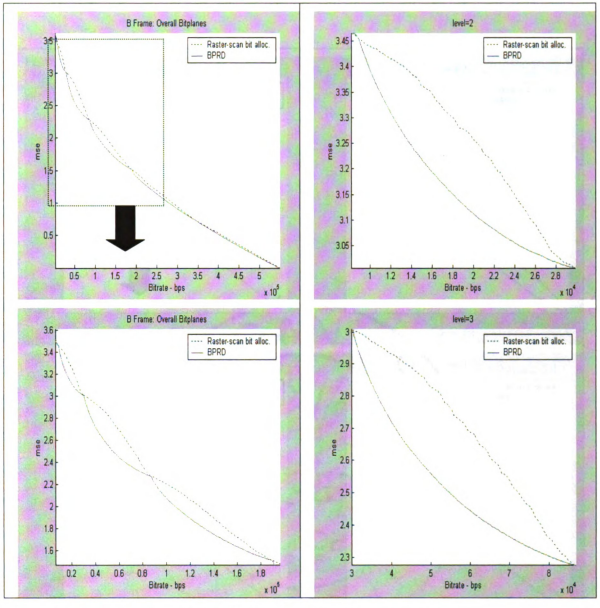


Figure 3-14 Foreman B frame Base layer=100k

apparent if the entire enhancement layer lies within the bandwidth range of 200 kbits/s to 2 Mbits/s. The other three curves in each figure hence, show the performance for these bitrates. It can be seen that the bitplanes that correspond to the specified enhancement layer bitrates vary between bitplane levels 0, 1, 2 and 3 for the Foreman sequence.

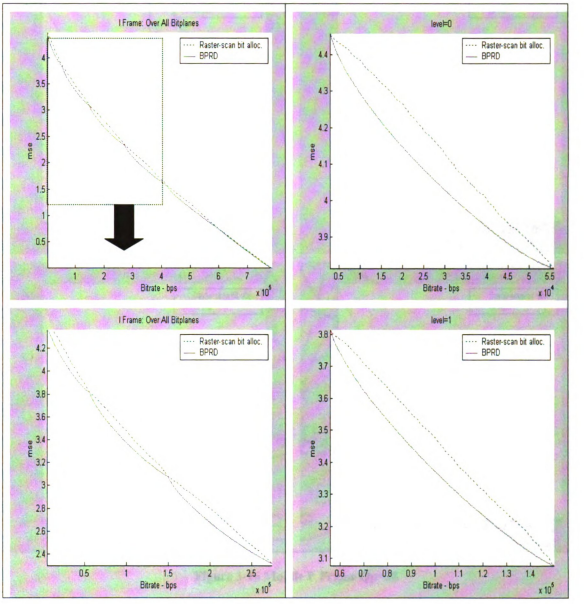


Figure 3-15 Mobile I frame for Qp=28

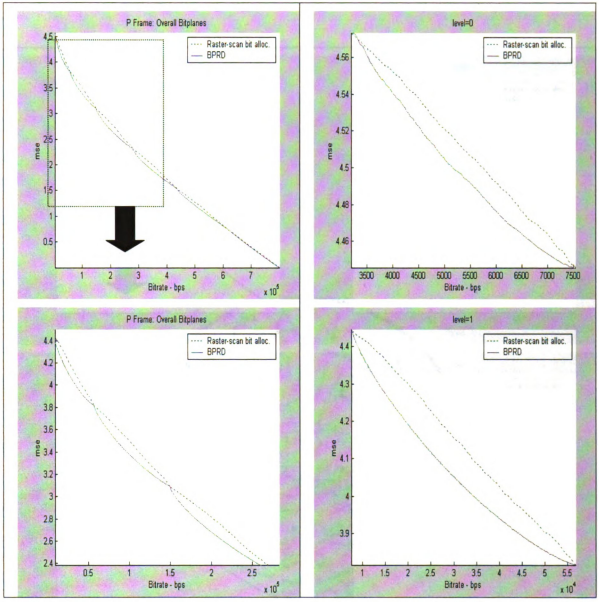


Figure 3-16 Mobile P Frame Qp=28

The last sequence that we analyze for the performance of the BPRD rate-control algorithm is the Mobile Calendar sequence. The Mobile sequence is a Class C sequence. It has high spatial detail and low amount of motion.

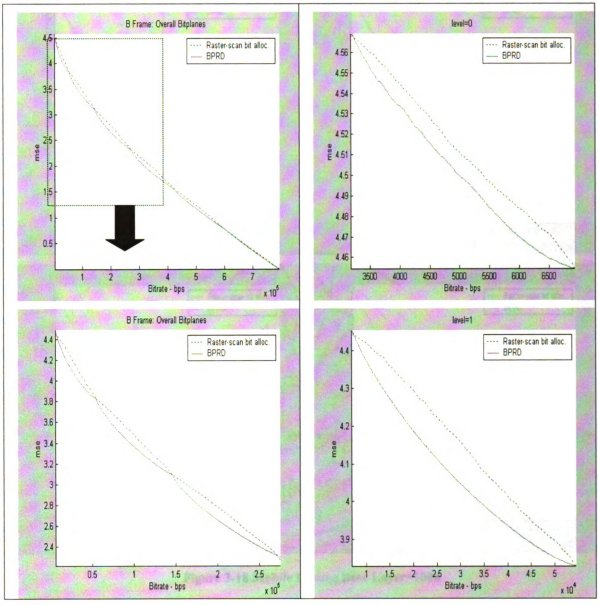


Figure 3-17 Mobile B Frame Qp=28

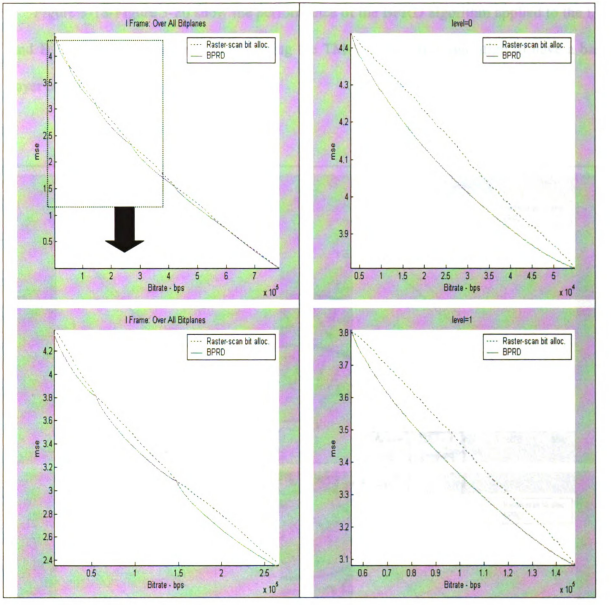


Figure 3-18 Mobile I frame Base Layer=100k

Figure 3-15-Figure 3-17 represent the results obtained for the I, P and B frames of the Mobile sequence coded using a constant $Q_p = 28$. This value of the quantization parameter Q_p corresponds approximately to a base layer coded at bitrate of 200 kbits/s.

Figure 3-18-Figure 3-20 show the performance of the BPRD algorithm applied to the I, P and B frames of the sequence coded using the TM-5 rate-control algorithm to have a base layer of 100 kbits/s.

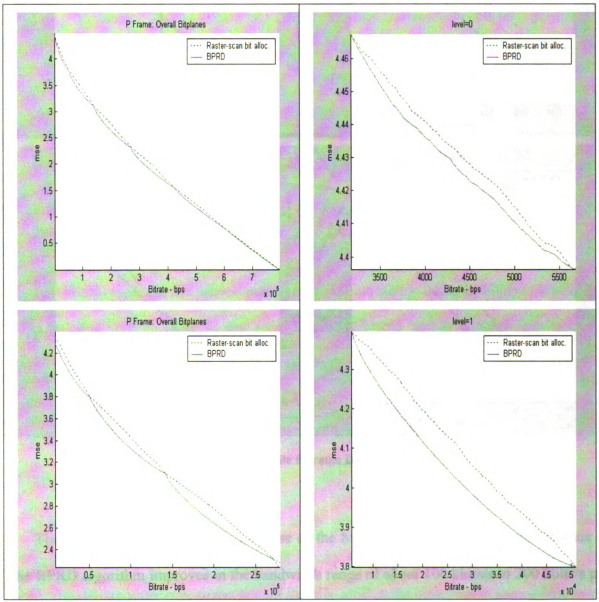


Figure 3-19 Mobile P Frame Base Layer=100k

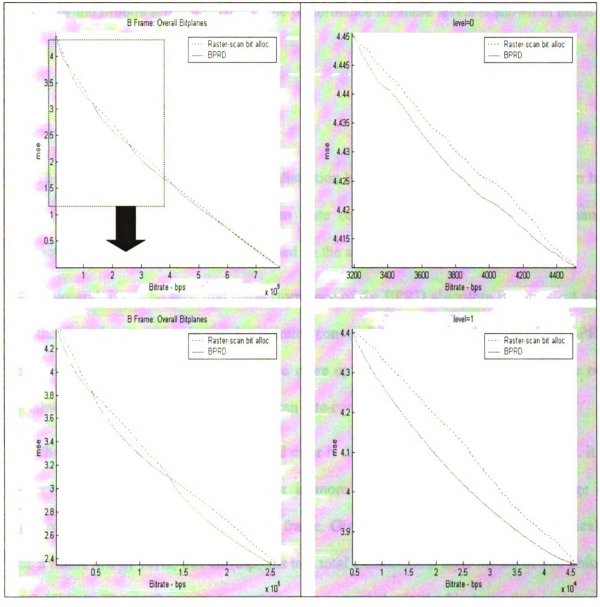


Figure 3-20 Mobile B frame Base layer=100k

The top-left curve indicates in the case of the Mobile sequence that the performance of the BPRD algorithm improves in the bandwidth range of about 100 kbits/s to 200 kbits/s per frame. This is equivalent to a total enhancement layer bandwidth of 1 Mbits/s to 2 Mbits/s for a frame rate of 10Hz. The bitplane levels that correspond to this bitrate range are levels 1

and 2 for the Mobile sequence. Hence, the performance for these levels is shown in detail in these figures.

3.4 Conclusions

The performance of the bitplane rate-distortion based rate-control (BPRD) algorithm has been presented thus far. We proceed to state certain conclusions on the performance characteristics of the BPRD algorithm based on the analysis of the presented results.

In general, it can be seen that the performance of the BPRD algorithm is consistent with the theoretical expectation in that it generated convex RD curves. Application of the BPRD to an individual bitplane level produces a more optimal rate-distortion curve than the one obtained by applying the existing raster-scan rate-control method.

The improvement in the mean-squared error is different for different bitplane levels. It is observed that this decrease in the error is more notable within a bandwidth range of approximately 50 kbits to 200 kbits per frame. Given that the encoded sequence typically has a frame rate of 10Hz, this corresponds to a total enhancement layer bitrate of 500 kbits/s to 2 Mbits/s. The performance improvement is noteworthy in the bitplanes that correspond to this target bitrate range.

The magnitude of the improvement in the mean-squared error is also differs from sequence to sequence. In the same target bitrate of 500 kbits/s to 2 Mbits/s, the BPRD algorithm gives a more significant lowering of the MSE in the Foreman sequence (A Class B sequence). The next best performance is for the Akiyo sequence (A Class A sequence). The

BPRD performs with only a fair amount of improvement for the Mobile sequence (A Class C sequence). This indicates that the performance of the algorithm depends to a large extent on the characteristics of the video sequence being encoded and transmitted.

The figures showing the performance of the BPRD algorithm on each individual bitplane, plotted together indicates that if the target bitrate should occur in the region of transition from one bitplane level to the next, the rate-distortion curve is locally non-optimal. Hence, in the next chapter, we propose an extension to the BPRD algorithm in order to find a rate-distortion performance curve that is optimal for all target bitrates.

4 Cross-Bitplane Rate-Distortion Based Rate-Control Algorithm

In this chapter, we extend the BPRD algorithm in an effort to optimize the rate-distortion curve at the region of transition from one bitplane to the next.

Our objective in the BPRD was to find the optimal subset $\mathfrak{M}_{opt(j)}$ from the set of all macroblocks M_j at bitplane level j in the transform (DCT) domain of a given frame. The cross-bitplane rate-distortion based rate-control (CBPRD) algorithm differs from the BPRD algorithm as follows. In this case, we search for the optimum subset $\mathfrak{M}_{opt}$ within the total set M. Here, M is the set of all *bitplane macroblocks* in the DCT domain of the frame under consideration. Consequently, the optimum subset $\mathfrak{M}_{opt}$ could include bitplane macroblocks that belong to different bitplane levels Figure 4-1.

Hence, under the CBPRD algorithm, we determine the optimal subset that minimizes the increase in distortion as measured across all possible macroblocks belonging to all possible bitplanes subject to a bitrate constraint:

$$\mathfrak{M}_{opt} = \arg\left\{ \min_{\substack{\mathfrak{M} \\ \sum_{m_i \in \mathfrak{M}} R_{m_i} \geq (R_{LL} - R_T)}} \left(\sum_{m_i \in \mathfrak{M}} \Delta D_{m_i} \right) \right\} \tag{4.1}$$

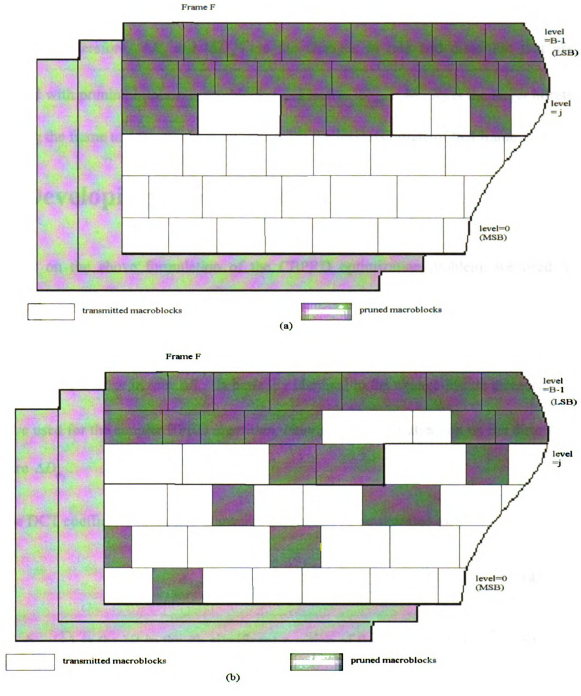


Figure 4-1 Pruning of macroblocks- width of the macroblocks represents the number of bits needed to encode that macroblock (a) BPRD – $\mathfrak{M}_{opt}$ is selected from bitplane level=j (b) CBPRD- $\mathfrak{M}_{opt}$ is selected from macroblocks on different bitplane levels

where m_i is bitplane i of macroblock m. In other words, m_i is an index to a *bitplane macroblock*. Therefore, ΔR_{m_i} and ΔD_{m_i} are the changes in rate and distortion that are associated with pruning bitplane macroblock m_i. Finally, $\mathfrak{M}_{opt} \subset \mathrm{M}$, R_T is the target bitrate for coding the frame under consideration, and R_{LL} is the lossless bitrate of that frame.

4.1 Developing The Algorithm

Based on the above formulation of the CBPRD optimization problem, we need to identify an efficient method for measuring the changes in rate and distortion measures for every bitplane macroblock m_i. As stated above, we desire to restrict the bitrate to a target bitrate R_T. The change in rate ΔR_{m_i} is basically identical to the change-in-rate measure that we have used for the simpler BPRD algorithm. Now we focus our attention on the distortion measure ΔD_{m_i}.

The DCT coefficient in block l of macroblock m can be represented as,

$$X_{l,m}(\bar{k}) = \sum_{i=0}^{B-1} 2^{(B-1)-i} b_{l,m,i}(\bar{k}) \tag{4.2}$$

where, $b_{l,m,i}(\bar{k})$ is the bit in the ith significant position of the coefficient $X_{l,m}(\bar{k})$ and B is the total number of bitplane levels needed to represent the block l of macroblock m.

The distortion associated with a block l of macroblock m is $\Delta D_{l,m}$ and the total distortion of macroblock m is ΔD_m. If the macroblock is considered at a bitplane level j such

that $0 \le j \le B-1$ and B is the total number of bitplanes in that frame, the block and macroblock distortions are represented by $\Delta D_{l,m_j}$ and ΔD_{m_j} respectively. These are denoted as follows,

$$\Delta D_{l,m_j} = \sum_{\bar{k}} \left[X_{l,m_j}(\bar{k}) - \hat{X}_{l,m_j}(\bar{k}) \right]^2 = \sum_{\bar{k}} \left[2^{(B-1)-j} \left(b_{l,m_j}(\bar{k}) - \hat{b}_{l,m_j}(\bar{k}) \right) \right]^2 \qquad (4.3)$$

and

$$\Delta D_{m_j} = \sum_{l} \Delta D_{l,m_j} \qquad (4.4)$$

Therefore,

$$\Delta D_{m_j} = \sum_{l} \sum_{\bar{k}} \left[X_{l,m_j}(\bar{k}) - \hat{X}_{l,m_j}(\bar{k}) \right]^2 = \sum_{l} \sum_{\bar{k}} \left[2^{(B-1)-i} \left(b_{l,m_j}(\bar{k}) - \hat{b}_{l,m_j}(\bar{k}) \right) \right]^2 \qquad (4.5)$$

Here again, $b_{l,m_j}(\bar{k})$ and $\hat{b}_{l,m_j}(\bar{k})$ can take values of 0 or 1. This implies that $b_{l,m_j}(\bar{k}) - \hat{b}_{l,m_j}(\bar{k})$ is also always 0 or 1. Therefore,

$$\Delta D_{m_j} = \sum_{l} \sum_{\bar{k}} \left[X_{l,m_j}(\bar{k}) - \hat{X}_{l,m_j}(\bar{k}) \right]^2 = \sum_{l} \sum_{\bar{k}} \left[2^{(B-1)-i} b_{l,m_j}(\bar{k}) \right]^2 \qquad (4.6)$$

Since ΔD_{m_j} is the accumulated distortion over certain specific bitplane levels and the distortion at every bitplane level is really the squared error, we shall refer to ΔD_{m_j} as the *accumulated squared error*. In the CBPRD, we shall use this accumulated squared error as the measure of distortion.

In equation (4.6), only the cases where $b_{l,m_j}(\bar{k}) = 1$ and $\hat{b}_{l,m_j}(\bar{k}) = 0$ contribute towards the total ΔD_{m_j}. Hence, operationally, the computation of ΔD_{m_j} can be achieved by

counting the number of ones in bitplane macroblock m_j, weighted by $2^{(B-1)-j}$. The weighting by $2^{(B-1)-j}$ indicates that the ones in the MSB bitplane (level j=0) are given the highest weight and those in the LSB bitplane (level $j = B-1$) are weighted the lowest. Consequently, if $N_1(m_j)$ is the number of ones in bitplane macroblock m_j, then ΔD_{m_j} can be expressed as follows:

$$\Delta D_{m_j} = \sum_{l}\sum_{\bar{k}}\left[2^{(B-1)-j} b_{l,m_j}(\bar{k})\right]^2 = 4^{(B-1)-j} N_1(m_j) \tag{4.7}$$

In the VLC tables developed for the FGS enhancement layer in [21], the number of bits required to code a block that consists entirely of zeros (or an all-zero block) is less than that required to code a block that contains one or more runs. Hence, converting a block to the all-zero state reduces the bitrate. This difference in the bitrate is the change in the rate $\Delta R_{l,m_j}$ due to converting (to zeros) the bitplane of block l of macroblock m at a given bitplane level=j. Therefore, the total number of bits reduced by pruning bitplane macroblock m_j:

$$\Delta R_{m_j} = \sum_{l} \Delta R_{l,m_j} \tag{4.8}$$

4.1.1 Description

Having defined the measures of rate and distortion, we proceed to outline the steps involved in the implementation of the CBPRD algorithm. Once again, it must be remembered that this algorithm operates on a frame-by-frame basis in an independent manner, and therefore does not extend across multiple frames.

Step 1: Start algorithm for frame *f*.

Step 2: Start from the bitplane with the lowest significance level, $i = B - 1$.

Step 3: Determine the change in rate $\left(\Delta R_{m_i}\right)$ and change in distortion $\left(\Delta D_{m_j}\right)$ for every bitplane macroblock m_j as described earlier in this section. Calculate the rate-distortion ratio $\left(\Delta D_{m_i} / \Delta R_{m_i}\right)$ for each macroblock.

Step 4: Repeat for all bitplane levels *i* until $i = 0$.

Step 5: Store references to all the macroblocks in an array and sort them in ascending order of the rate-distortion ratio.

Step 6: Convert the macroblock referenced at the bottom of the sorted array to all-zero. Move reference one step at a time up the sorted order.

Step 7: Check if the target bitrate has been reached by this conversion.

Step 8: If the target bitrate has not been reached, go to *Step 6*.

Step 9: If the target bitrate has been reached, re-encode the modified DCT coefficients using variable-length coding.

Step 10: Repeat for all frames *f*.

A single flag indicates the *all-zero* conversion in the CBPRD. The choice of macroblocks is made to reduce the overhead in recreating the compressed bitstream. We now proceed to look at the experimental analysis of the CBPRD.

4.2 Experimental Results

For the experimental analysis of the cross-bitplane rate-distortion based rate-control (CBPRD) algorithm, we define the same test bed as was done for the BPRD. The video format used is the CIF format. The test sequences used are the Akiyo (Class A sequence), Foreman (Class B sequence) and Mobile (Class C) sequence.

The performance of the CBPRD algorithm on these test-sequences is analyzed for the cases when the sequences are coded with and without application of rate-control to the base layer bitstream. When rate-control is not applied, the base layer is encoded using a constant quantization parameter Qp. The Qp values used are 4, 6 and 8 for the Akiyo sequence and 10, 15 and 28 for the Foreman and Mobile sequences. The other case considered is application of the TM-5 rate control algorithm to the base layer. In this case, the three sequences were encoded to constrain the base layer bitrate to 100 kbits/s, 200 kbits/s and 300 kbits/s.

The results are presented below in order of the test-sequences. As before, we start with the results of applying the CBPRD algorithm to the I, P and B frames of the Akiyo sequence followed by the results for the I, P and B frames of each of the Foreman and Mobile sequences. For all these cases, the figure showing the results consists of three performance curves.

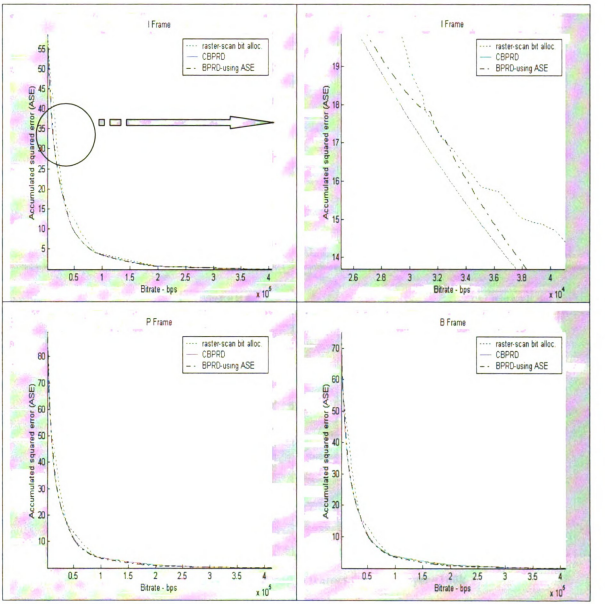


Figure 4-2 Akiyo I, P and B frames for Qp=8

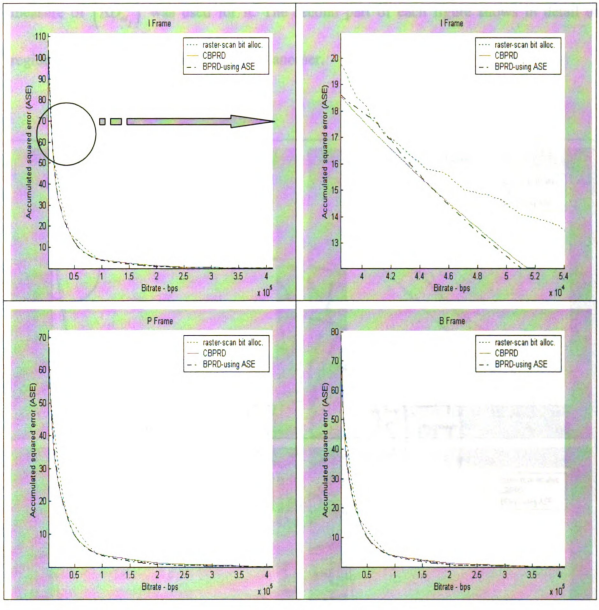


Figure 4-3 Akiyo I, P and B frames for Base layer=100k

The first curve is the performance of the existing raster-scan rate-control method. The second curve indicates the performance of the CBPRD algorithm. The third curve that has

been plotted gives an insight into the performance of the BPRD algorithm, if the distortion measure of $\left(\Delta D_{m_j}\right)$ was used for it. The second part of each figure shows in detail one region of transition from one bitplane to another.

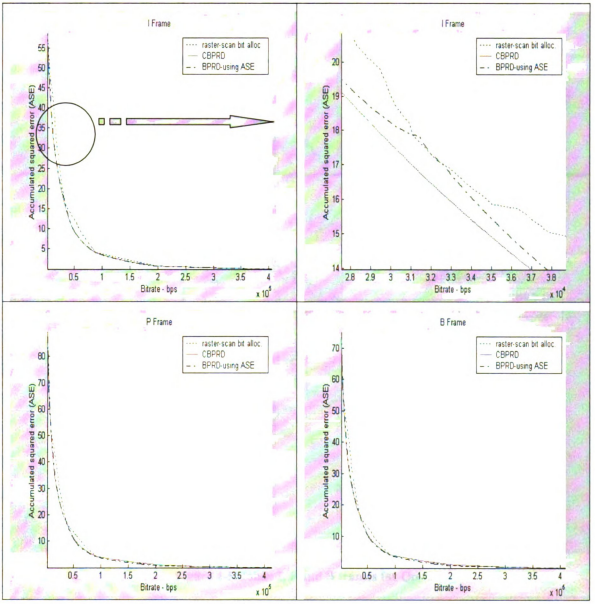


Figure 4-4 Foreman I, P and B frames for Qp=28

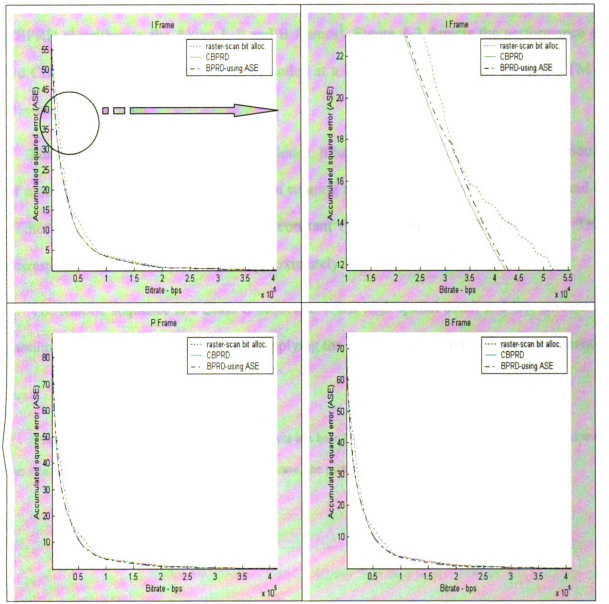


Figure 4-5Mobile I, P and B frames for Qp=28

It can be seen from the figures, that the CBPRD algorithm performs more optimally in these regions than the BPRD would, if the same measure of distortion were to be used for both the algorithms.

The Akiyo sequence is a Class A sequence. Figure 4-2 shows the results of applying the CBPRD algorithm to the Akiyo I, P and B frames[2]. Figure 4-3 depicts the performance of the CBPRD algorithm for the sequence coded at a base layer of 100 kbits using the TM-5 rate-control algorithm.

The Foreman sequence is Class B sequence. It has medium spatial detail and low amount of motion. Figure 4-4 shows the results of running the CBPRD algorithm on the I, P and B frames for the sequence coded using a constant Qp value of 28 in the base layer. This corresponds to a base layer bitrate of approximately 100 kbits/s.

The Mobile sequence is a Class C sequence. It has high spatial detail and low amount of motion. Figure 4-5 shows the results of applying the CBPRD algorithm to the enhancement

---

[2] The square error numbers shown in these plots are based on averaging the accumulated square error over the size of the picture under consideration. In this case, the picture size is 352x288 pixels.

The Mobile sequence is a Class C sequence. It has high spatial detail and low amount of motion. Figure 4-5 shows the results of applying the CBPRD algorithm to the enhancement

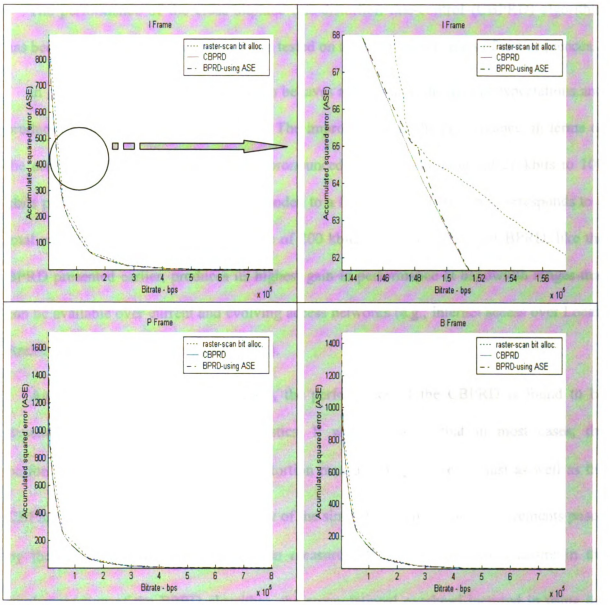


Figure 4-6 Mobile I, P and B frames with Base layer=100k

4.3 Conclusions

The performance of the cross-bitplane rate-distortion rate-control (CBPRD) algorithm has been presented. The results have been tested on three different classes of test-sequences.

In general, the CBPRD algorithm also behaves according to theoretical expectations and produces a convex rate-distortion curve. The improvement in the performance, in terms of the accumulated squared error is more pronounced in the bitrate range of 20 kbits to 100 kbits per frame. Considering sequences coded to a frame rate of 10Hz, this corresponds to a total enhancement layer bandwidth range of 200 kbits/s to 1 Mbits/s. The CBPRD, like the BPRD presented earlier, produces its highest gain in performance at bandwidth ranges that can be available over current and evolving access networks (e.g., Internet access over Local-Area-Networks, cable modems, DSL etc.).

Again, as in the case of the BPRD, the performance of the CBPRD is found to be dependent on the sequence characteristics. It was also seen that in most cases, the performance of the BPRD using the distortion measure ΔD_{m_j} performs just as well as the CBPRD itself. This indicates that in view of the simpler implementation requirements posed by the BPRD, calculating the distortion measure ΔD_{m_j} and using this measure in the implementation of the BPRD algorithm, the computational overhead incurred in using the CBPRD can be reduced.

Consequently, for scalable video rate-control systems that desire to achieve optimum rate-distortion performance while maintaining the lowest possible computational

complexity, the BPRD algorithm provides a good option. As shown above, the BPRD algorithm virtually performs as well as the more optimum and more complex CBPRD.

5 Summary Of Conclusions

In this thesis, we have developed and analyzed the performance of two rate-distortion based algorithms for scalable video rate control. These algorithms are tailored for the recently developed Fine-Granularity-Scalable (FGS) video coding methods such as the one adopted by the MPEG-4 video standard.

The first rate-distortion based rate-control algorithm operates on a bitplane-by-bitplane basis within an FGS compressed bitstream. We refer to this algorithm as the bitplane RD algorithm (BPRD). A more general RD algorithm that operates across all possible bitplanes of an FGS coded stream is subsequently presented. We refer to the second algorithm as the cross-bitplane RD (CBPRD) algorithm.

The distortion measures used for these two algorithms are different. The distortion measures selected in both cases are based on the popular square-error criterion. One of our primary objectives has been the ease of computation of the distortion measure. We have shown that a simple count of ones found in the bitplanes of the lossless (original) video in the transform domain corresponds directly to a square-error measure.

We have shown that the RD algorithms developed using the specified distortion measures gave a better rate-distortion performance than the existing raster-scan rate-control method that is currently implemented in the MPEG-4 FGS verification model. It was also

observed that the region of greatest improvement corresponds to a bandwidth range that can be available over current and evolving Internet access networks such as LANs and DSL.

The calculation of the rate and distortion can be done off-line and the rate-distortion ratio for every macroblock can be stored prior to transmission. This implies that implementing these RD based algorithms in real-time calls for very little computational overhead during the real-time streaming process of the desired video.

5.1 Future Work

One extension of the work done in this thesis is to incorporate the algorithm into existing decoder systems for the MPEG-4 FGS standard coded video. This will verify that the theoretical improvements observed in the transform (DCT) domain presented in this thesis can also be observed in the pixel domain.

In the case of image and video systems, determining the best measure of distortion is a challenge. The distortion measures used in this thesis were designed to reduce the computational complexity, while remaining as close as possible to the actual mean-squared error in the transform domain. Thus, another area of extending the work done in this thesis is to explore the possibility of finding better measures of distortion that can correlate better with subjective improvements as perceived by the human visual system. This could include incorporating different weights for the distortion associated with the different DCT coefficients in the transform domain.

Finally, we are currently investigating the correlation among the three key measures used in the development of the proposed algorithms: change-in-rate ΔR, change-in-distortion ΔD, and their ratio $\Delta D / \Delta R$. This correlation study could lead to an efficient and optimum estimate of the RD ratio ($\Delta D / \Delta R$) based on measuring only one of the two parameters (ΔD, ΔR). This approach could be useful for systems that can easily measure (or have access to) one of the parameters but not the other.

References

[1] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall Information and System Sciences Series, pp. 476-560, 1989.

[2] A.K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349-384, 1981.

[3] A. Ortega and K. Ramachandran, "Rate distortion methods for image and video compression," *IEEE Signal Proc. Magazine*, pp. 23- 50, Nov. 1998.

[4] A. Puri and R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 1, pp. 351-361, 1991.

[5] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. Journal,* vol.27, pp.379-423, 1948.

[6] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *IRE National Convention Record, Part 4,* pp. 142-163, 1959.

[7] C. Gonzales and E. Viscito, "Flexibly scalable digital video coding," *Signal Processing: Image Communication*, vol. 5, Feb. 1993.

[8] CCITT Recommendation H.120, "Codecs for videoconferencing using primary digital group transmission," Geneva, 1989.

[9] CCITT Recommendation H.261, "Video codec for audiovisual services at p x 64 kbit/s," Geneva, 1990.

[10] CCITT SG XV, "Draft revision of recommendation H.261," Document 572, Mar. 1990.

[11] D. J. Le Gall, "The MPEG video compression algorithm," *Signal Processing: Image Communication*, vol. 4, pp. 129-140, 1992.

[12] D. Lee, "New work item proposal: JPEG 2000 image coding system," ISO/IEC JTC1/SC29/WG1 N390, 1996.

[13] E. Viscito and C. Gonzales, "A video compression algorithm with adaptive bit allocation and quantization," *Proc. SPIE,* 1605, pp. 58-72, 1991.

[14] F.Ling, W.Li, and H.Sun, "Bitplane coding of DCT coefficients for image and video compression," in *Proc. SPIE Visual Communications and Image Processing (VCIP)*, San Jose, CA, Jan.25-27, 1999.

[15] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electronics*, vol. 38, pp. 18-34, 1992.

[16] H. G. Musmann, M. Hotter and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," in *Signal Processing: Image Communication*, vol. 1, pp. 117-138, 1989.

[17] H. Radha and Y. Chen, "Fine-granular-scalable video for packet networks" in *Packet Video '99*, New York, Apr. 1999.

[18] H. Radha, M. van der Schaar, Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, Mar. 2001.

[19] ISO/IEC 11172-2, "Information technology- coding of moving pictures and associated audio for digital storage media at up to about 1. Mbits/s- video," Geneva, 1993.

[20] ISO/IEC 14496-2, "Information technology- coding of audio-visual objects: visual," Committee Draft, ISO/IEC JTC1/SC29/WG11, N2202, Mar. 1998.

[21] ISO/IEC JTC/SC 29/WG 11 N3515, "MPEG-4 video verification model version 17.0".

[22] ISO/IEC JTC1/SC29/WG11, "MPEG-4 version 2 visual working draft rev 3.0," N2212, Mar. 1998.

[23] J. De Lameillieure and R. Schafer, "MPEG2 image coding for digital TV," *Fernseh und Kino Technik*, 48. Jahrgang, pp. 99-107, Mar. 1994 (in German).

[24] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann, 1996.

[25] L. Stenger, "Digital coding of television signals—CCIR activities for standardization," *Signal Processing: Image Communication*, vol.1, pp. 29-43, 1989.

[26] M. Hauben, "Behind the net- the untold history of the ARPANET," http://www.dei.isep.ipp.pt/docs/arpa.html

[27] M. Nelson, *The Data Compression Book*, M&T Books, 1995.

[28] MPEG-4 FAQ, ISO/IEC JTC1/SC29/WG11, July 1997. http://drogo.cselt.stet.it/mpeg/faq/faq_mpeg-4.htm

[29] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," *Proc. IEEE,* Oct. 1993.

[30] N. Wakamiya, M. Miyabayashi, M. Murata and H. Miyahara, "MPEG-4 video transfer with TCP-friendly rate control," *Proc. 4th IEEE Intl. Conf. on MMNS*, 2216, pp. 29-42, 2001.

[31] R. C. Nicol and N. Mukawa, "Motion video coding in CCITT SG XV – the coded picture format," in *Conference Record of IEEE Global Telecommunications Conference and Exhibition—Communications for the Information Age*, Hollywood, vol. 2, pp. 992-996, Dec. 1988.

[32] R. J. Clark, *Transform Coding of Images*, Academic Press, Orlando, Florida, 1985.

[33] R. Schifer and T. Sikora, "Digital video coding standards and their role in video communications," in *Proc. IEEE*, pp. 907-924, vol. 83, Jun. 1995.

[34] Scalability for stereo video coding, http://vr.kjist.ac.kr/~3D/Research/Stereo/scale_video.html

[35] T. Berger and L. D. Davidson, "Advances in source coding," in *CISM- Courses and Lectures*, no. 166, Udine 1975.

[36] T. Ebrahimi, "MPEG-4 video verification model 8.0," ISO/IEC JTC1/SC29/WG11 MPEG-97 Doc No. W1796, July 1997.

[37] T. Sikora and L. Chiariglione, "The MPEG-4 video standard and its potential for future multimedia applications," *Proc. IEEE ISCAS Conf.*, Hong Kong, June 1997.

[38] T. Sikora, " The MPEG-4 video standard verification model," *IEEE Trans. CSVT*, vol. 7, Feb. 1997.

[39] "Test model 5," ISO/IEC JTC1/SC29/WG11/No. 400, MPEG93/457, Apr. 1993.

[40] Testing *Ad Hoc* Group, "JPEG2000 testing results." ISO/IEC JTC1J/SC29/WG1/N705, Nov. 1997.

[41] V. Paxson, "End-to-end internet packet dynamics," in *Proc. ACM SIGCOMM*, vol. 27, pp.13-52, Oct. 1997.

[42] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. CSVT*, vol. 11, Mar. 2001.

[43] W. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, 1994.

[44] W.H. Chen and W.K. Pratt, "Scene adaptive coder," *IEEE Trans. Communications,* pp. 225-232, 1984.

[45] W.Li, F.Ling and H.Sun, "BitPlane coding of DCT coefficients," ISO/IEC JTC1/SC29/WG11, MPEG97/M2691. Oct. 1997.