USC–SIPI REPORT #308

Video Bit-Rate Control with Spline Approximated Rate-Distortion Characteristics

by

Liang-Jin Lin

May 1997

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Dedication

To my parents

Acknowledgments

I would like to express my deep appreciation to my advisor, Prof. Antonio Ortega, for his valuable advice, munificent help, and friendship during the essential stage of my study. I also would like to express my deep gratitude to Prof. C.-C. Jay Kuo, for his generous support, constant encouragement and friendship throughout the study at USC. Without their help in all matters, this work would never have been accomplished.

I am grateful to Prof. Wlodek Proskurowski for his great mathematics classes and for finding valuable time and efforts to serve on my dissertation committee. I also thank Prof. Alexander A. Sawchuk and Prof. Zhen Zhang for their suggestions and encouragement at my qualifying exam.

I would like to thank my colleagues at Signal and Image Processing Institute and Integrated Media Systems Center, Dr. Kwo-Jyr Wong, Dr. Chun-Hsiung Chuang, Dr. Li-Chien Lin, Dr. Yu-Chuan Lin, Dr. Jin Li, Dr. Junavit Chalidabhongse, Dr. Po-Yuen Chang, Dr. Wei-Feng Chen, Xia Wan, Victor Liang, Yung-Kai Lai, Mei-Yin Shen, Ioannis Katsavounidis, Brian Beard, Mike Wang, Te-Chung Yang, Jiankun Li, Asha Vellaikal and other members of the group for their help and friendship for last few years.

My deepest gratitude goes to my parents who have devoted themselves to my education with constant love and enthusiasm since my childhood.

Contents

List of Figures

List of Tables

Abstract

Digital video's increased popularity has been driven to a large extent by a flurry of recently proposed international standards (MPEG-1, MPEG-2, H.263, etc.). In order to transmit compressed videos over a communication channel, it is required to control the compression in the encoder such that the output data rate can be fit within the channel constraints. The rate control scheme, which plays an important role for improving and stabilizing the decoding and play-back quality, is not defined in the standard and thus different strategies can be implemented in each encoder design. The control scheme defined in MPEG Test Model 5 provides a solution with very light computational overhead. However, the results are not always good for any given video sequence and channel rates. It is also difficult to adjust parameters to improve the quality without the help of some trial and error encoding tests. Moreover, it does not control the buffer to prevent buffer overflow. Several rate-distortion (R-D) based techniques have been proposed. The new approaches solved all these problems and generate better and more stable results. However, these approaches are complex because they require the R-D characteristics of the input data to be measured before making quantization assignment decisions.

In this research, we show how the complexity of computing the R-D data can

be reduced without reducing too much the performance of the optimization procedure. We propose three stages which provide successive reductions in complexity. In the first stage, we propose an algorithm based on penalty functions and iterative gradient search. The computational complexity is reduced because we only need to evaluate rate and distortion functions along the search path, which is much less than the requirement in other R-D based methods (e.g., trellis-based approaches). Although the algorithm only converges to a local optimum solution, our experiments show that it is close to the global optimum solution.

The second stage of the research focuses on the approximation of rate and distortion functions, which can greatly reduce the complexity of R-D based rate control techniques. Previous work was mainly based on statistical models, for which it is difficult to accurately determine the model parameters, and where the model error may be too large to be useful for our R-D optimized rate control algorithm. Therefore, we propose an approximation method based on computing a few R-D points and interpolating the remaining points using spline functions. The inter-frame dependency of R-D functions is also considered and modeled by linear-constant functions. MPEG encoding tests show that, by using the proposed approximated R-D functions within our gradient-based rate control scheme, the results are very close to the ones based on the original R-D data, with only about 15% to 20% of computations.

In the final stage, we propose a fast algorithm suitable for real-time encoding. The R-D optimized approach is still used in the algorithm, except that the R-D data for those un-coded frames are predicted from the coded frames. Properties of human visual system (HVS) are also used to enhance the visual quality. The experimental results shows that better and more stable quality can be achieved

by our algorithm, especially when the channel rate is low (e.g., CIF format at 192 to 256 kbits per second). All our algorithms and encoding results are compatible with standard MPEG decoders, hence the resulting encoded sequences can be used by any MPEG decoder to achieve the improved quality.

Chapter 1

Introduction

1.1 Significance of the Research

The use of digital techniques for recording and transmitting signals has been known to have many desirable features such as robustness and flexibility. Recently, after remarkable success of digital technology for audio (e.g., CD), digital video has gained more and more attention. Compared to audio, the major problem for video is its excessive amount of data, which makes efficient data compression indispensable. The effort for defining a video compression standard started in 1988, when a study group named *Moving Picture Expert Group*, or MPEG, was formed [28]. The original goal for this standard was to put video programs in the same media used by digital audio, the compact disk. Hence the data rate was set to be the same as digital audio, about 1.5 megabits/sec. This standard, known as MPEG-1 [21], was finalized at the end of 1992. Many hardware and software codecs has become available since then. Now, after quick development of multimedia computers, the standard has been widely used to encode the video

clips in interactive multimedia programs. However, its original goal as a standard for CD-Videos, which store up to 74 minutes of continuous video in VHS quality on a compact disk, has not been achieved successfully, because, when encoding at the standard bit-rate and resolution, its play-time is not long enough to fit a full-length movie and its quality is not good enough to supersede the VCR. Therefore, there are not many dedicated CD-Video players available, and most of playback devices have been built for use with personal computers, where significant CPU power is used and large amount of main memory space is allocated as decoder buffer. The follow-up standard, MPEG-2 [22], which covers a wide range of applications from current broadcast-quality video to HDTV, was also finalized at the end of 1994. The first application of the new standard is for satellite broadcasting (e.g., DirectTV) to deliver TV programs with better quality, more channels, and better protection from unauthorized viewing. Another application is on the Digital Video Disc (DVD), which features at least ten times the capacity of the current standard CD, and can store up to 133 minutes of video at 4.69 megabits/sec, and is expected to replace current VCR tapes and Laser Disks for delivering programs and movies.

The superior features of digital video can not be realized without a good video compressor, because playback quality is mostly determined by the encoder, where compression settings have to be determined for video frames such that the output rate of bit stream can be fitted into a data channel. Beside the quality for each individual frame, another fact that also deteriorates the quality is that the system sometimes may lose the synchronization and begin dropping pictures periodically, which makes the playback un-smooth. And again, the causes of the synchronization problem are usually not in the decoder, but in the encoder. This is because the compressed video stream is in variable-bit-rate (VBR) format due

to the use of entropy coding, while the satellite broadcasting and CD-ROM playback are using constant-bit-rate (CBR) channels. (For CD-ROM playback case, the CD-ROM drive is a slow device and data has to be read out continuously at a fixed rate, so it is considered as a CBR channel.) Hence, memory buffers have to be put between the encoder and CBR-channel to smooth out the variation, and also between the CBR-channel and the decoder to restore the original VBR format, as shown in Fig. 1.1. The diagram shows the case for real-time encoding and decoding. The loss of synchronization is usually caused by buffer overflowing. To prevent this from happening, the bit-stream production rate in the encoder should be regulated by a bit-rate control algorithm. The goal of *bit-rate control* is to, given a video sequence, a buffer size, and a channel rate, determine the quantization settings in the encoder, such that the buffer does not overflow. These constraints stem from the synchronous operation of encoder and decoder with a constant delay in the system. A detailed description of the connection between buffering and delay constraints can be found in [47]. Additional requirements are usually added to improve the video quality given the rate and buffer constraints, such as optimizing the quality of each picture frame, while keeping the quality difference between consecutive frames small to potentially maintain constant quality in playback. In the CD-ROM or DVD case, although the encoding can be done off-line, the bit-rate still needs to be controlled correctly to prevent the decoder buffer from overflowing during the real-time playback. Since the compression can be done off-line and only has to be done once, more computing power will be available for the encoding, hence more sophisticated pre-analysis and optimization algorithms will become feasible to improve and maintain the quality, which is essential for video program or movie publishers to control the quality of their

encoding.

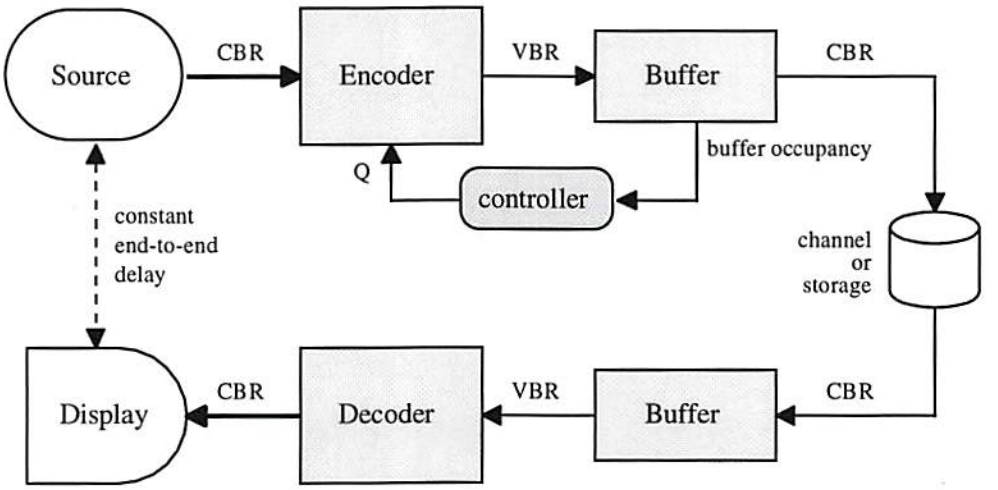


Figure 1.1: Video encoder and decoder

Another type of application is for real-time video communication, or video conferencing. The first standard for this purpose is H.261 [23], which is designed to be used over $p \times 64$ kbits/sec channels (e.g., ISDN). A newer standard for low-bit-rate channels (lower than 64 kbits/sec such as V.34 modem over telephone line), H.263 [24], is also proposed and now under draft phase. These standards share many similar building blocks with MPEG, and again, the bit-rate control scheme plays an important role for improving and stabilizing the quality of playback. Because of low-delay requirements in these applications, the buffer size should be keep small to reduce buffer-delay, which makes the bit-rate control even more essential.

Besides the CBR encoding, recently, the variable-bit-rate (VBR) encoding, which is useful for transmitting videos over statistically multiplexed network (e.g., ATM, Ethernet, or other Internet links), has also gained more and more attention. Potential applications include video on demand, video conference over Internet,

wireless video phone, etc. It has been shown that when multiplexing several video streams into a single communication link, properly designed VBR encoding usually can achieve better qualities for all parties over CBR encoding, hence effectively increasing the capacity of the channel [37, 55, 48, 47, 45, 6, 20].

Therefore, research aimed at improving video compression quality and efficiency over a large variety of channel conditions is very important. There are two major research topics in the area of bit-rate control, namely, (i) developing more effective and generic optimization coding control strategies for any given channel conditions, and (ii) developing more accurate and efficient models for estimating the rate-distortion characteristics of video sources. The second topic is necessary to make the optimization procedures successful and computationally realizable. In this dissertation, we address both topics. Although our first focus has been mainly based on the video compression standards (MPEG-1, MPEG-2, H.261, H.263) over CBR channels, the algorithms and models are generic and, with simple modifications, are applicable to more general scenarios such as VBR channels, proprietary algorithms and some of emerging MPEG-4 schemes.

1.2 Video Compression and Rate Control

In this dissertation, we use the MPEG standard, which we now introduce briefly in this section, as a basis for our rate control algorithm and our experiments.

1.2.1 MPEG Encoding

We show in Fig. 1.2 a typical block diagram of an MPEG encoder. The input picture X_n is segmented into blocks of 16×16 pixels, called the *macroblocks*, which

are the basic encoding units in MPEG. Each macroblock can be a non-intra block or an intra block. A non-intra block does not depend on any other blocks, so the feedback path in the diagram is not used, and the value of pixels go directly to the DCT compression unit. For an intra block, prediction is first formed by using pixels in the reference frame, at the same location with a displacement by a motion vector (also known as the motion compensated prediction). Then, predicted values are subtracted from the pixels in the current block and the results are called prediction residue. The residue Y_n is further segmented into blocks of 8×8 pixels and transformed to the DCT domain. The resulting DCT coefficients are then quantized by a set of uniform quantizers, whose step sizes are determined by a quantization table and quantization scales known as *mquant*'s. By definition in the standard, the quantization step size is proportional to the value of *mquant*, except for the DC coefficients of intra-block, whose quantization step-size is fixed. The quantized coefficients are then encoded by an entropy coder, where the zigzag scanned zero-run length and Huffman coding is used. Note that during the encoding, the quantization error, or distortion d, can be calculated in the quantization stage, and the code-length r can be calculated in the entropy coding stage, as indicated in the figure.

The intra/non-intra selection strategy is not defined in the standard, but there are some constraints according to different types of picture frames. Three frame types are defined in MPEG, known as I, P, and B. In I (intra) frames, all the macroblocks should be coded as intra block, i.e., they are compressed without using the information from any other frames. In P (predicted) frames, each macroblock can be coded as intra block or non-intra block. In B (bi-directionally interpolated) frames, each macroblock can be coded as an intra-block, with forward prediction

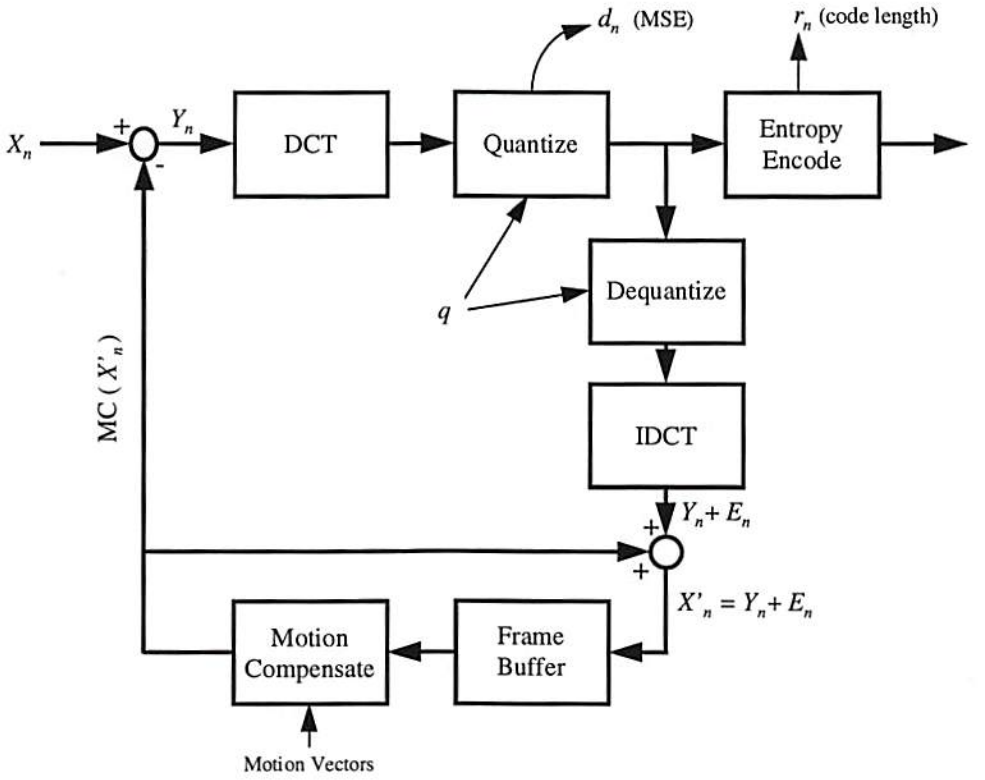


Figure 1.2: Typical encoder block diagram

only, with backward prediction only, or as a bi-directionally interpolated block. Fig. 1.3 shows a typical frame configuration and their dependencies. The set of pictures is called group of pictures (GOP). Note that in order to decode a B frame, the information from its future P or I frame is required, therefore those P or I frames have to be encoded and transmitted before their previous B frames. More details of MPEG encoding can be found in [28, 19, 38] and in the standard documents [21, 22].

In Fig. 1.2, two parameters have to be supplied for each macroblock in order to complete the encoding, namely, motion vector and quantization scale. The motion vector is calculated by a motion estimation algorithm, and the quantization scale is determined by a bit-rate control algorithm. These two algorithms are not specified

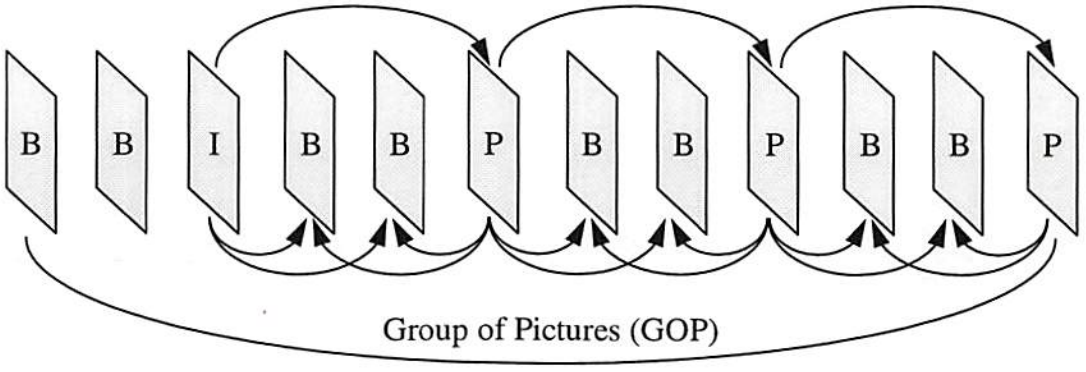


Figure 1.3: Typical frame configuration in a group of pictures (GOP)

in the standard and are often considered as key components to the encoding speed and quality. The study of the motion estimation algorithm is out the scope of this dissertation. We will focus on the bit-rate control algorithm in this research.

1.2.2 Bit-Rate Control

In MPEG, the value of *mquant* assigned for each macroblock determines the quantization step-size, and hence controls the rate-distortion trade-off. The permissible values for *mquant* are integers from 1 to 31. The objective of bit-rate control is to determine *mquant* for each macroblock to keep the output bit-rate within the rate and buffer constraints while maintaining good and stable quality.

Depending upon the types of communication channel to be used, the bit-rate control can be designed for variable-bit-rate (VBR) encoding or constant-bit-rate (CBR) encoding. Earlier VBR encoders, e.g. [39], simply use a constant value for *mquant*'s to encode the entire video. Because there is no control over bit-rate, the only application is for computer-based playback where very large buffer (main memory) and fast channel (hard drive) are available. The second type of VBR

encoder is targeted for the ATM networks, where it is often necessary to regulate the bit-rate in order to comply with some constraints imposed by the network [2]. The last type, CBR encoder, can be used for CD-ROM or DVD devices or other dedicated constant rate channels such as ISDN or satellite broadcast channels. In this dissertation, we concentrate in the case of constant bit rate (CBR) encoding. Transmission over variable bit rate (VBR) channels may also require a rate control strategy to allow synchronous operation of encoder and decoder [47, 6, 20] and some of our results could be extended to VBR transmission.

Fig. 1.4 shows a VBR to CBR conversion through buffering. The memory buffer is used to smooth out the bit-rate variations. These bit-rate variations not only depend on the image contents, but also depend on different frame types, i.e., at a similar quantization setting and quality level an I frame generates more bits than a P or B frame. A buffer has also to be put at the decoder so that the video can playback synchronously, since the information received at constant rate is read by the decoder at variable rate [47]. The rate control mechanism is required for determining the quantization settings for each block such that the encoder buffer does not overflow, or equivalently, the decoder buffer does not underflow.

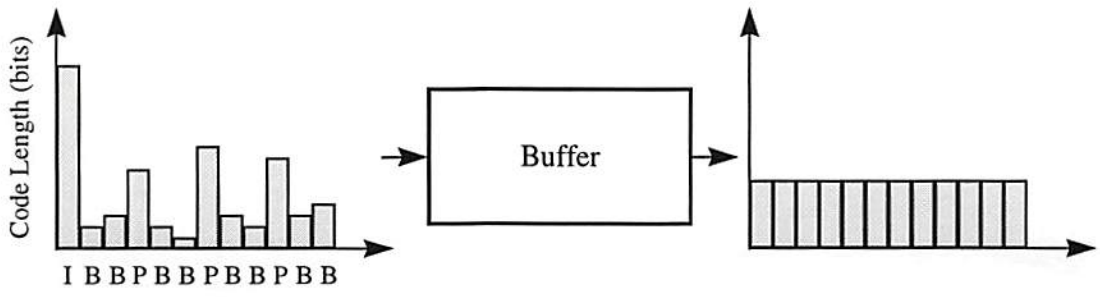


Figure 1.4: VBR to CBR conversion

We can classify bit-rate control schemes into two major classes depending

on whether they are based on prediction or pre-analysis. In *predictive* control schemes, rate allocation decisions are based on current information such as the buffer state or the expected rate for future blocks. Examples include direct buffer-state feedback methods where the buffer occupancy determines the quantization setting [4]. These methods may suffer in performance if the assumptions, which may be based on a particular type of sequence or scene, do not hold. Moreover, most predictive methods suffer from degradation at scene changes since they operate based on a model that is no longer valid. The advantages of these approaches are low computational complexity and low delay at the bit-rate control stage, thus making them suitable for real-time communication environments such as video conferencing.

The second class of bit-rate control schemes is based on *pre-analysis*, in which the choice of *mquant* can depend on future macroblocks or frames which are read and cached in the encoder. The schemes are usually called *delay-decision control schemes* because the pre-analysis causes additional delay in the rate control stage. The encoding quality is usually better than that of predictive scheme. Some methods have been proposed which are based on a single-frame pre-analysis, including the works in [26, 18, 59, 13]. Other schemes are based on rate-distortion optimization techniques. Their goal is to meet the constraint of overflow prevention while maximizing the video quality. Methods based on Lagrangian optimization [58, 8, 46, 31, 27] or dynamic programming [43] have been considered. These methods typically rely on much more intensive pre-analysis, usually over an entire GOP, to measure the rate-distortion characteristics. If frame dependencies are taken into account [46] the complexity can become very high making some of these methods only suitable for off-line encoding.

A detailed review of the various classes of bit-rate control schemes will be presented in Chapter 2.

1.3 Outline of the Dissertation

In this dissertation, our objective is to develop new rate control techniques which give good and stable quality, with reasonable computation complexity for practical applications. In Chapter 2, we survey several previous works on the rate control problem, ranging from low-complexity MPEG Test Model 5 (TM5) algorithm to high-complexity pre-analysis-based optimization algorithms. Although low-complexity methods such as TM5 can give a reasonable result for some video sequences at some range of channel rates, they usually require "manual tweaking" on the control parameters to achieve good result for a specific selection of video sequence and channel rate. Therefore, we consider techniques motivated by rate-distortion criteria within a delayed-decision pre-analysis optimization framework. With the optimization mechanism, our techniques can automatically adapt to any given video sequence and channel rate without any manual intervention.

Because the major disadvantage of these optimization-based approaches is high computational cost, our first step is to reduce computation requirements to a level where practical implementation is possible. To achieve this goal, we develop a new buffer control technique which uses penalty functions and iterative gradient search techniques for solving the optimization problem. The computations are reduced because, instead of measuring R-D data on all possible quantization settings, we only measure the data along the search path. Note that because the cost function is not perfectly smooth, the solution may be trapped in a local optimum point

during search process. However, we have verified that the solution is in general still close to the optimal point achievable through exhaustive search. A detailed description of the algorithm and experimental results is presented in Chapter 3.

In the second step, we introduce a new approximation model to reduce computation complexity by avoiding the need to measure the R-D data on all possible settings. The models we propose are better suited to rate-distortion optimization in realistic video coding scenarios, because they (i) make relatively few assumptions on the shape of the R-D characteristics and are thus suited when operating with a small number of quantizers and (ii) take into account the dependencies typical of video coding. These models are based on computing a few R-D points and interpolating the remaining points using spline functions. Compared to other models, e.g. those based on exponential R-D functions, ours provide better approximation in terms of model errors. The price to pay for the increased accuracy is a somewhat higher complexity. We apply our models to the gradient-based method and show that, with only about 15% of computation cost, we can achieve performance close to that obtained with the actual R-D data. The approximation model and its application are presented in Chapter 4.

In the third step, based on our rate-distortion approximation scheme, we propose a new fast algorithm with only one frame of delay. The algorithm combines the simplicity of a predictive control scheme, such as the one defined in the MPEG Test Model 5, and our rate-distortion optimization approach, using the predicted rate-distortion characteristics. Simulation results show that both the PSNR and the stability of the quality are improved. We then re-configure the algorithm for optimizing the visual quality instead of PSNR, and achieve significant improvement on the quality over MPEG Test Model 5, especially when the channel rate is

limited. In Chapter 5, we introduce this fast bit-rate control scheme and present several experimental results.

Finally, concluding remarks and future extensions are stated in Chapter 6.

1.4 Summary of Contributions

The following contributions have been made in this dissertation.

- A thorough survey has been made for the bit-rate control algorithms, from fast predictive control schemes to computation intensive deterministic rate-distortion optimal control schemes.

- We have studied and analyzed the MPEG TM5 algorithm, focusing on its assumptions, effects of parameter choices, and performance.

- We have proposed a new bit-rate control algorithm based on gradient search, and achieved nearly optimum solution with less computation complexity compared to other deterministic optimum control schemes.

- We have proposed a new approximation model for the rate and distortion characteristics. In the model, the intra-frame R-D data is approximated by spline interpolation functions, and inter-frame dependency is modeled by a linear-constant function. Extensive model compliance tests have been carried out.

- Based on the original gradient-based bit-rate control algorithm and the new proposed R-D approximation model, we have proposed a new algorithm which achieves a solution similar to the results from the original method, with only about 15% of computations.

- We have proposed a new fast bit-rate control algorithm, with only one frame delayed for pre-analysis. Compared to TM5, the method gives more stable quality with only small amount of computation overhead (several additional quantization and encoding operations for each frame).

- Based on our fast algorithm, we propose a scheme to optimize the visual quality using pre-filtering and block-classification procedures. The side-by-side playback comparison with TM5 shows that our encoding algorithm gives better and more stable quality, especially at lower bit-rate where the impact of rate control scheme is more significant.

- We have implemented all our new methods in standard MPEG format. All the encoded video were compatible with any standard MPEG player. Furthermore, they could be easily adapted to be used within a H.261, H.263, and other video coding standards.

Chapter 2

Bit-Rate Control Techniques

2.1 Introduction

As mentioned in the previous chapter, we can classify bit rate control schemes into two major classes depending on whether they are based on prediction or pre-analysis. In *predictive* control schemes, the quantization settings for each macroblock only depends on the current macroblock and all the previous encoded macroblocks. In the second class, *delayed-decision pre-analysis* control schemes, the quantization settings can be dependent on future macroblocks or future frames. Depending on the number of frames delayed for pre-analysis, these delayed-decision schemes can be further categorized into either single or multiple (usually an entire group of picture) frame delay.

In this chapter, we review the existing bit-rate control schemes. In Section 2.2 we review general ideas of predictive rate control scheme. In Section 2.3 we review and analyze MPEG Test Model 5 (TM5) algorithm, which was originally presented in MPEG Test Model documents [51, 40]. The results from TM5 algorithm will be

used extensively throughout this dissertation as references[1]. In Section 2.4 we review some delayed-decision control schemes which are based on the rate-distortion optimization. And finally, in Section 2.5 we review and evaluate previous works on the approximation models for rate-distortion characteristics.

2.2 Predictive Bit-Rate Control Scheme

As introduced in the previous chapter, for MPEG encoding, the basic compression unit is a 16×16 macroblock and the bit-rate is controlled by the value of *mquant* assigned to each macroblock. The possible value for *mquant* is an integer number between 1 and 31. The simplest way to control the compression is to use buffer occupancy level to determine the value of *mquant* for the next block, as shown in Fig. 2.1. The method is also known as direct buffer-state feedback scheme.

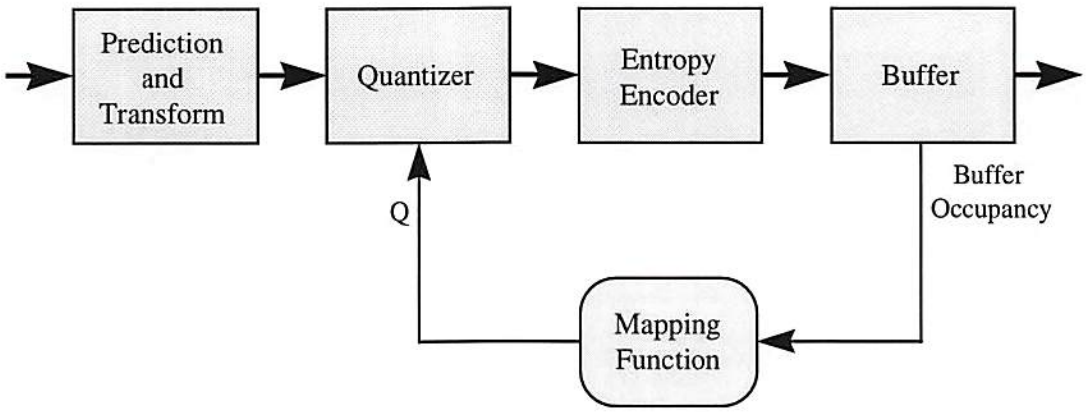


Figure 2.1: Direct Buffer-State Feedback Control Scheme

In our formulation, the following notations are used.

[1]Although better methods are available in commercial systems, TM5 is representative of fast and simple rate control approaches

R : channel rate in bits per second.

F : frame rate in frames per second.

M_b : total number of macroblocks in a frame.

r_b : channel rate in bits per block, i.e., $r_b = R/(F \cdot M_b)$.

i : index to a macroblock.

q_i : value of *mquant* assigned to the block i, where

$$q_i \in \{1, 2, \ldots, 31\}, \quad \forall i = 1 \ldots M_b.$$

r_i : number of bits generated by the block i. Note that the value of r_i depends on q_i, and may also depends on the reference frame if it is a non-intra block.

$b(i)$: buffer occupancy (in bits) after the i^{th} block is coded.

The buffer occupancy after block i is coded can be calculated by

$$b(0) = b_0, \quad b(i) = \max\left(b(i-1) + r_i - r_b, 0\right), \tag{2.1}$$

where b_0 represents the initial buffer occupancy. Note that the max function is used in (2.1) because a negative value of buffer occupancy does not have any physical meaning, and we can simply add padding bits into the bit stream when the buffer is underflowing. A typical controller which maps the buffer occupancy

into quantization setting q could be

$$q_{i+1} = \begin{cases} 1, & b(i) \leq 0.2b_{max}, \\ 50 \cdot b(i)/b_{max} - 9, & 0.2b_{max} < b(i) < 0.8b_{max}, \\ 31, & b(i) \geq 0.8b_{max}, \end{cases} \tag{2.2}$$

where b_{max} is the buffer size, and q_{i+1} will be rounded to an integer and used as *mquant* for the next block. The mapping function is shown in Fig. 2.2. This mapping provides a negative feedback mechanism because when the buffer occupancy becomes higher, the *mquant* also becomes larger and thus reduces the output bit-rate and decreases the buffer occupancy. Other types of mapping functions were also proposed in previous works. For example, in [10], an S-shaped mapping function was used in the controller. Note that in the mapping (2.2), the effective control range for the buffer occupancy is confined to be within 20% to 80% of buffer fullness, so that the probability of buffer overflow and underflow can be lower. If the buffer is underflowing, we can simply add padding bits. On the other hand, if the buffer is overflowing, the only thing we can do is to drop blocks, which will severely degrade the output quality. Because the control is done at macroblock level, thus can be labeled a *Local Control* scheme.

There are many disadvantages in the above feedback control scheme. The first problem is caused by the use of *Local Control*, which may cause quality varying within a frame. The problem can be solved if we can control bit-rate at frame level by using a single control parameter for each frame (also known as *Global Control*), and leave the value of *mquant*'s in a frame to be either constant or adaptively adjusted in a way (independent of rate control) aimed at improving visual quality. However, this approach usually increases quality variation between consecutive

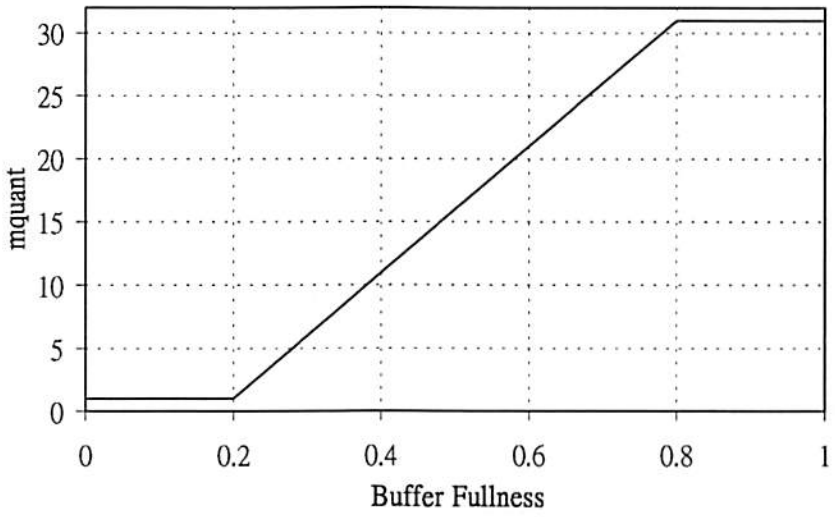


Figure 2.2: A typical mapping function which maps a buffer fullness to a quantization scale. The buffer fullness of 1 means full buffer, or b_{max} in the formulation.

frames and also increases the probability of buffer underflow and overflow, thus making the control of bit-rate more difficult. Hence, it is not practical unless a rate-distortion optimization control approach is introduced (see Section 2.4). Another approach for reducing quality variations is to add a control stage to damp the change of *mquant*, for example, by defining

$$q_i' = q_i + \alpha \cdot (q_{i-1} - q_i), \quad 0 < \alpha < 1, \tag{2.3}$$

and using q_i' as *mquant* instead of q_i. By slowing down the variation of *mquant*, we not only get a more uniform quality, but also reduce the coding overhead for *mquant*, which requires 5 bits for each change in value (see [42], which addresses optimum allocation when the overhead is important). Unfortunately, the new controller will also decrease the control on rate and increase the probability of buffer underflow and overflow. The work in [4] is intended to cope with this

problem. Other works suggested a non-linear mapping function in the controller, which was adopted by MPEG-2 standard by providing an optional mapping from *mquant* (now between 1 and 62) to quantization step-size (still 31 different choices) [22].

However, the most obvious disadvantage in the above approach is that the different properties of I, P, B frames are not considered. The feedback mechanism tends to decrease bit-rate for I frames and increase bit-rate for B frames, which usually decrease the overall quality. The first practical control algorithm for MPEG which takes the difference of I, P, B into account is presented in an MPEG test model specification and is often referred to as Test Model 5 (TM5) algorithm [40]. The algorithm is described in the next section.

2.3 MPEG Test Model 5 Algorithm

The TM5 algorithm is based on the direct buffer-state feedback method, with modifications to take into account the difference between frame types (I, P, B) and keep the quality at a similar level for all frames. The algorithm uses a two-step control approach, with a *Global Control* which calculates a control parameter q (quantization scale) at frame level, followed by a *Local Control* which determines *mquant* for each macroblock based on both q and buffer fullness. To explain how the algorithm controls the "frame quality", we use mean squared error (MSE) for quality measurement, defined as

$$d = \frac{1}{H \cdot W} \cdot \sum_{x=1}^{H} \sum_{y=1}^{W} (v(x,y) - \hat{v}(x,y))^2 , \tag{2.4}$$

where $v(i,j)$ is the pixel value of the original image, $\hat{v}(i,j)$ is the pixel value of the quantized image, and H and W are the height and width of the image respectively.

2.3.1 Assumptions on R-D Characteristics

The following assumptions on the rate and distortion characteristics of frames were made to keep the algorithm simple.

1. The distortion d increases linearly with quantization scales. The rate of increase (or slope) for different frame types is different, as shown in Fig. 2.3.

2. In order to achieve the same distortion for different frame types, the values of quantization scales, denoted as q_I, q_P, q_B for each frame-type respectively, should be kept at a constant ratio, i.e.,

$$\frac{q_I}{1} = \frac{q_P}{k_P} = \frac{q_B}{k_B} \tag{2.5}$$

   where k_P and k_B (known as q-ratio parameters) are constants.

3. The total code-length of a frame, denoted as r, is inversely proportional to the distortion d, i.e., $r \cdot d = constant$.

Based on the first and third assumptions, we have $r \cdot q = constant$. Thus we can define a frame activity (or complexity) for each frame type as

$$\begin{aligned} A_I &= r_I \cdot q_I, \\ A_P &= r_P \cdot q_P / k_P, \\ A_B &= r_B \cdot q_B / k_B, \end{aligned} \tag{2.6}$$

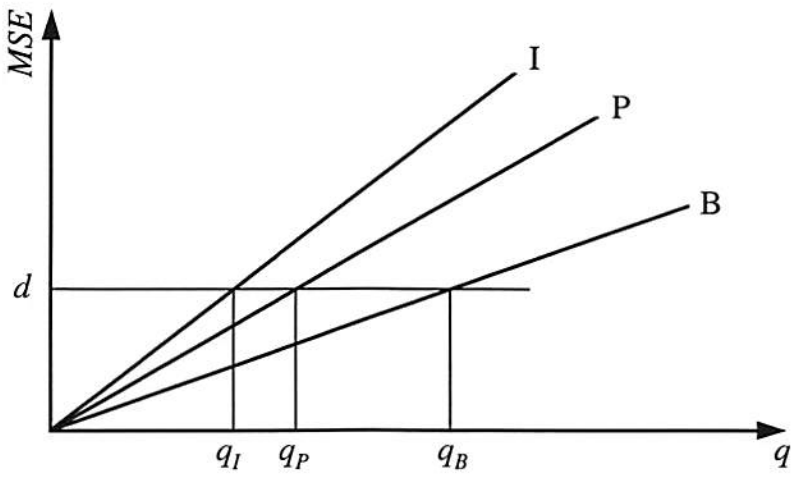


Figure 2.3: Ideal distortion-quantization curves of I, P, B frames in TM5. At the same distortion level, the ratios between q_I, q_P, and q_B are fixed.

where r_I, r_P, and r_B are the frame code-length for each frame type, and k_P and k_B are used compensate the difference between I, P, B frame types, based on the second assumption. Note that, because the value of *mquant* can be changed for different macroblocks, the values of quantization scales (q_I, q_P, q_B) in (2.5) and (2.6) have to be defined as the average of *mquant*'s over all macroblocks in a frame.

2.3.2 Control Procedures

Global Bit Allocation

In TM5, a Group of Pictures (GOP) is considered as a basic unit for rate control. The bit allocation is made in such a way that the number of bits assigned for a frame is proportional to its activity measure in (2.6), and the total number of bits assigned to the GOP meets the bit budget. Note that in the predictive encoding model, the frame activity of a future frame is not known until it is coded, and

thus the activity of the previous frame of the same type is used instead. Let the numbers of I, P, B frames in a GOP be N_I, N_P, N_B, respectively. For example, for a GOP with structure like that of Fig. 1.3, $N_I = 1$, $N_P = 3$, and $N_B = 8$. The total bit budget, denoted as B, is

$$B = B_0 + (N_I + N_P + N_B) \cdot \frac{R}{F}, \tag{2.7}$$

where B_0 is the number of bits left (or over-used if it is negative) from the previous GOP, R is channel rate in bits per second, and F is frame rate in frames per second. Then, the target number of bits for each frame type, denoted as T_I, T_P, and T_B respectively, can be derived by solving the following set of equations,

$$\begin{cases} T_I/A_I = T_P/A_P = T_B/A_B \\ N_I T_I + N_P T_P + N_B T_B = B. \end{cases} \tag{2.8}$$

The solution is

$$T_I = \frac{B}{N_I + N_P \cdot A_P/A_I + N_B \cdot A_B/A_I}, \tag{2.9}$$

$$T_P = \frac{B}{N_i \cdot A_I/A_P + N_P + N_B \cdot A_B/A_P}, \tag{2.10}$$

$$T_B = \frac{B}{N_i \cdot A_I/A_B + N_P \cdot A_P/A_B + N_B}. \tag{2.11}$$

The target bit-budget of a frame is then used for the next two steps described in the following sub-sections. The actual code-length generated by the frame is usually different from the target bit-budget, and thus after a frame is coded, the total bit budget B is updated by subtracting the bits actually consumed by the coded frame, and one of the N_I, N_P, N_B is decreased by one (depending on the

frame type of the coded frame) to reflect the number of frames left for encoding in the GOP. Then, equation (2.8) is re-calculated to get a new target bit-budget for the next frame. Also, because the activity is measured from the previously encoded frames, the values of A_I, A_P, or A_B in (2.8) are also measured and updated for future use, based on the new available data after a frame of a given type has been encoded.

Local Rate Control

After the bit-budget for a frame is determined, a direct buffer-state feedback technique is used to monitor the code-length and determine quantization scale for each macroblock. Suppose there are M_b macroblocks in a frame, the number of bits per macroblock is

$$\begin{aligned} c_I &= T_I/M_b, \\ c_P &= T_P/M_b, \\ c_B &= T_B/M_b, \end{aligned} \tag{2.12}$$

where the subscript denotes the frame type. To take into account the difference between frame types, three virtual buffers are used. By using i ($i = 1 \ldots M_b$) to index the macroblocks, the occupancy for the three virtual buffers can be derived as

$$\begin{aligned} b_I(i) &= b_I(i-1) + r_i - c_I, \\ b_P(i) &= b_P(i-1) + r_i - c_P, \\ b_B(i) &= b_B(i-1) + r_i - c_B, \end{aligned} \tag{2.13}$$

where r_i is the code-length of i^{th} macroblock, and the initial buffer occupancy is carried over from the previous frame. Then, a simple scaling is used to map the

buffer occupancy into q (global quantization scale),

$$q_i = \begin{cases} 31 \cdot g \cdot b_I(i) \\ 31 \cdot g \cdot b_P(i) \\ 31 \cdot g \cdot b_B(i) \end{cases} \tag{2.14}$$

where the parameter g determines the gain of controller.

Adaptive Quantization

The global quantization scale q_i from the rate control stage is ready to be used to quantize the macroblock, but in TM5 it is weighted further by a "block activity" factor. The block variance of the original pixel values is used to measure the block activity, denoted as x_i for the i^{th} block. To reduce the weighting for macroblocks that contain an edge, for which higher quality is usually preferred, the variances of the four 8×8 blocks, denoted as σ^2_{imn}, $m, n = 1, 2$, are calculated separately, and then the minimum among them is picked, so that the block activity is

$$x_i = 1 + \min(\sigma^2_{i11}, \sigma^2_{i12}, \sigma^2_{i21}, \sigma^2_{i22}). \tag{2.15}$$

Then, the scaling factor is calculated by the following non-linear mapping,

$$s_i = \frac{2 \cdot x_i + \overline{x}}{x_i + 2 \cdot \overline{x}}, \tag{2.16}$$

where $\overline{x}$ is the average of x_i over an entire frame, calculated from the previous frame. After this mapping, the range of s_i is confined in between 0.5 and 2, as shown in Fig. 2.4. Finally, the value $s_i \cdot q_i$ (rounded to the nearest integer between

1 and 31) is used as *mquant* to quantize the macroblock. The purpose of this adaptive quantization scheme is to re-distribute the bits from high activity blocks to low activity blocks, so that bits for all blocks are somewhat equalized, which effectively reduce chance of buffer overflow. This mapping also gives more bits to the smoother areas in the picture frame and less bits to the texture areas, thus tending to preserve the quality of smoother areas.

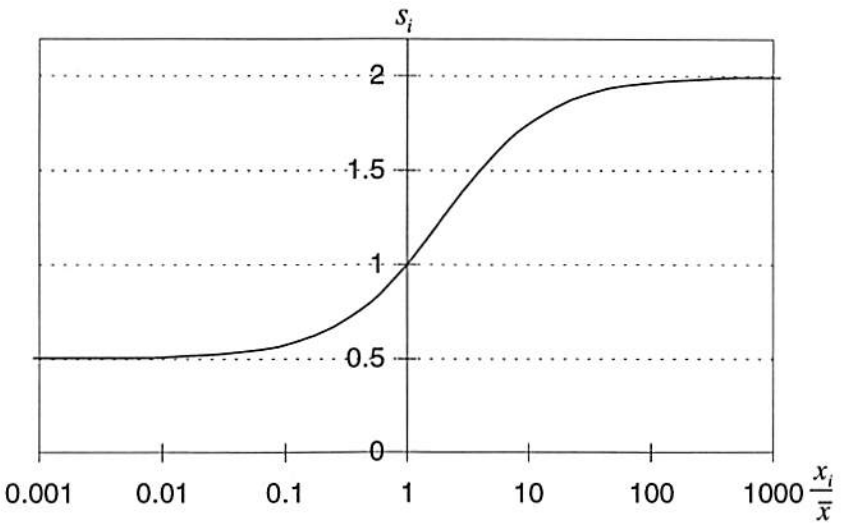


Figure 2.4: Nonlinear mapping function for adaptive quantization.

2.3.3 Problems in TM5 Algorithm

There are several problems in the TM5 algorithm:

1. The basic assumptions on the R-D characteristics may not be met in practice. To examine the compliance of the assumptions, we repeatedly encode the same frame with different type (I, P, and B) using the 31 quantization scales (all the macroblocks use the same quantization in each encoding).

The resulting distortion curves are shown in Fig. 2.5, and the frame activities in Fig. 2.6. From the figures, we see that the Football sequence (high activity) is closer to the assumptions than the Miss America sequence (low activity). In general, the error can be somewhat large for particular video sequences.

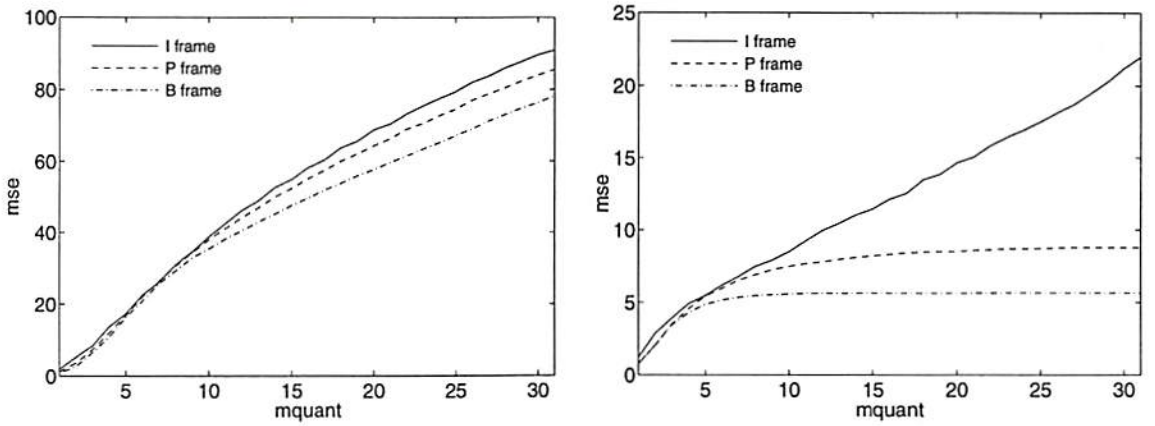


Figure 2.5: Distortion as function of q for the three frame type, measured from (a) Football and (b) Miss America.

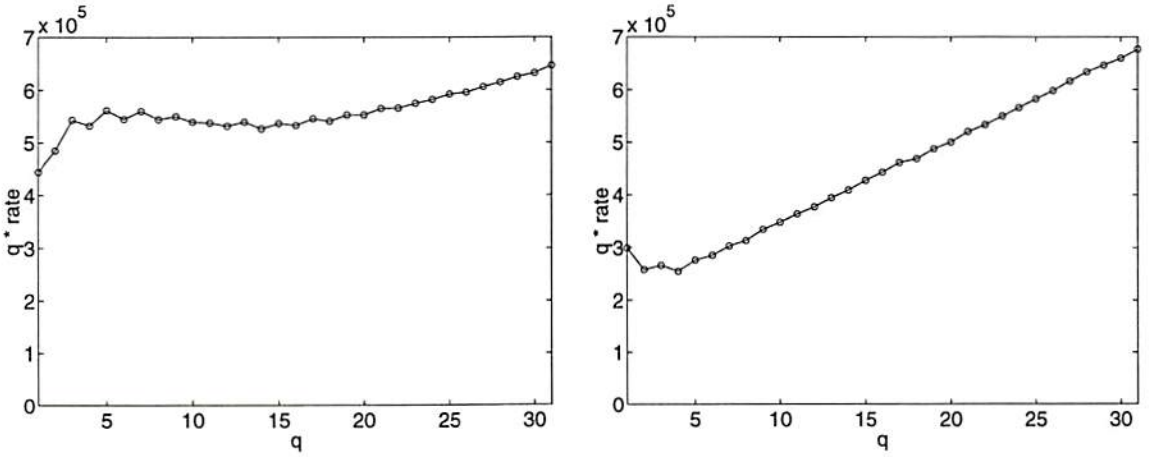


Figure 2.6: "Activity" of a frame in (a) Football and (b) Miss America, computed by multiplying the code-length and the quantization scales.

2. There are several parameters that need to be specified in the algorithm. The ratio parameters, k_P and k_B in (2.5), specify the relative slope of the distortion-quantization function between different frame type. It directly affects the relative output quality between different frame types. The values suggested in [40] are $k_P = 1.0$ and $k_B = 1.4$, which are also used in the Software Simulation Group's codec implementation [41]. To evaluate the effect of the parameters, we repeatedly encode the Football sequence with many different settings of k_P and k_B. The average distortion (MSE) of a GOP is shown in Fig. 2.7. The figure shows that the minimum distortion actually occurs around $k_P = 1.2$ and $k_B = 1.7$. Other experiments show that the optimum values of k_P and k_B depend on the content of video. One possible enhancement is to find a better way to determine the ratio parameter (k_P, k_B) rather then just use constant values, as suggested in [25]. Another parameter, the controller gain g in (2.14), determines the mapping between the buffer occupancy and the quantization scales. A larger value can reduce the transition time during scene changes but could also make the system become unstable, i.e., oscillating between good and bad quality. In TM5 document, the suggested value of g is $0.5 \times (R/F)$. Although the values of k_P, k_B and g can be properly tuned for a specific video clip in order to achieve good and stable results, it is difficult to find and use a single set of parameters to achieve good results for all video sequences.

3. The algorithm does not include a mechanism to prevent buffer overflow. As shown in (2.13), three virtual buffers, one for each frame type, are used instead of a single real buffer. Hence, the real buffer fullness is not monitored and the control may fail, especially when the buffer size is small.

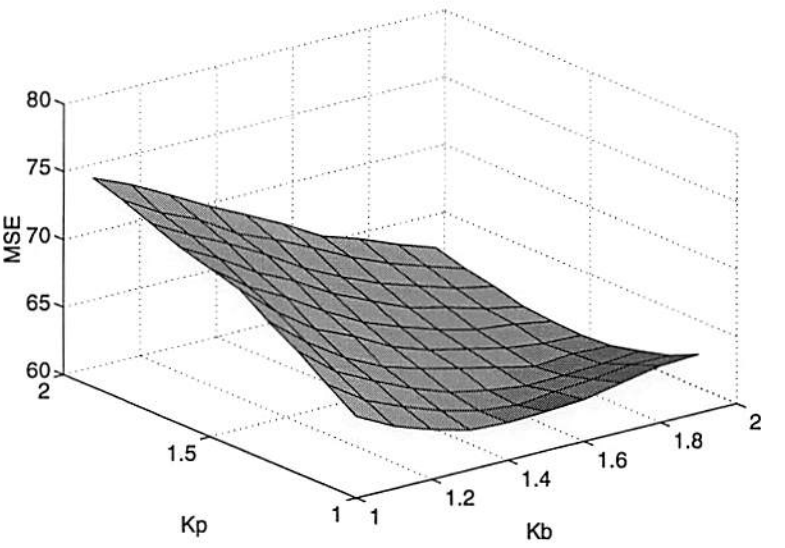


Figure 2.7: Average distortion (MSE) as a function of k_P and k_B, measured from a GOP of Football sequence. The minimum distortion is occurred at $k_P = 1.2$ and $k_B = 1.7$

4. The algorithm cannot handle scene changes well. This is due to the "predictive" requirement that the encoding cannot depend on the future blocks, and thus the average block activity in (2.16) and frame activities in (2.6) are all measured from previous frames, which might be quite different from current frame, especially after a scene change.

Many control schemes have been proposed to cope with these problems. Some of them are based on pre-analyzing the rate-distortion characteristics of the input sequence and then calculating the best choice of quantization settings which optimize the quality. One important feature of these schemes is that they make very few assumptions on the rate-distortion characteristics, therefore, in theory, they can be applied equally well to virtually all video sequences and channel rates. In the next section, we review some previous work on these schemes.

2.4 Delayed-Decision Bit-Rate Control

In delayed-decision rate control schemes, one or several frames are read and analyzed before the decision for q is made. These schemes usually give better quality through the use of pre-analysis. Most of the recent research activities on bit-rate control belong to this category. In most cases the pre-analysis is based either on one frame or on a GOP.

For the single-frame delay case, the TM5 algorithm can be directly improved by measuring the frame and block activities from the current frame before it is encoded, rather than from previously coded frames. Another TM5-based method proposed in [26] uses a constant q to quantize and encode all the macroblocks in the frame to get a bit-usage profile, and then uses the profile during the real encoding phase to get better prediction and control on the rate. Another method proposed in [18] measures the entropy and uses it to predict the bits at macroblock level to, again, get better control on the rate.

A common problem of these algorithms is that, the quality, or the distortion, is not measured and controlled explicitly. To cope with this problem, another class of rate control methods have been proposed which take the distortion into account. The goal of these methods is to determine the best achievable quality for a given channel rate and buffer size, if the entire video sequence or part of its "future frames" are known in advance. Fig. 2.8 shows a typical block diagram of these rate-distortion based control techniques. The solution found by these techniques can be used as a benchmark for other rate control algorithms. In this section, we present surveys of these techniques.

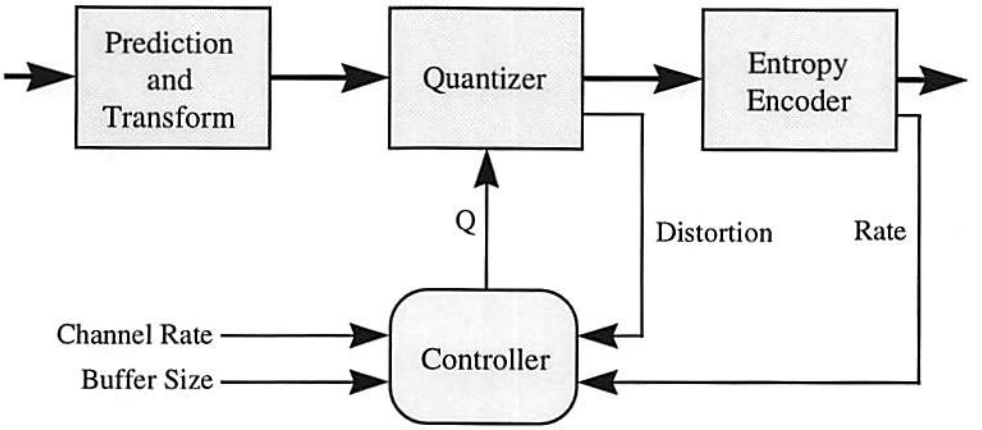


Figure 2.8: Rate-distortion based control scheme

2.4.1 Lagrange Multiplier Techniques

We first consider the case where the *blocks* are assumed to be coded independently, and where the buffer size is infinite (no buffer constraints). The independence of quantization means that the quantization setting for a given block does not affect any of the other blocks. The size of a "block" can be a macroblock in a Local Control case, or a frame in a Global Control case. In this section, we use Local Control as an example, and use the same notations as defined in Section 2.2, i.e., R denotes the channel rate in bits per second, F denotes the frame rate in frames per second, M_b denotes the total number of blocks in a frame, i denotes the index to a macroblock, and q_i denotes the value of *mquant* assigned to the block i where

$$q_i \in \{1, 2, \ldots, 31\}, \quad \forall i = 1 \ldots M_b.$$

The rate and distortion of a block are then defined as below.

$r_i(q_i)$: number of bits generated for block i, when q_i is used to encode the block. We refer to this as a "rate-quantization function".

$d_i(q_i)$: distortion introduced by the quantization of block i, when q_i is used to encode the block. Any kind of meaningful distortion measure can be used here, including the MSE defined in (2.4) We refer to this as a "distortion-quantization function".

The rate-control problem is to determine q_i, $i = 1, 2, \ldots, M_b$, for all blocks such that the total distortion is minimized and the total number of bits does not exceed the total budget $B = R/F$ (assuming that the basic control unit is a frame).

Formulation 2.1 *Independent Block Coding without Buffer Constraints: Determine* q_i, $i = 1, 2, \ldots, M_b$ *to*

$$\textit{minimize} \sum_{i=1}^{M_b} d_i(q_i), \quad \textit{subject to} \quad \sum_{i=1}^{M_b} r_i(q_i) \leq B. \tag{2.17}$$

The problem can be solved using the standard method of Lagrange multipliers, based on the following theorem.

Theorem 2.1 *Lagrange Multiplier Method*

If a set of q_i^*, $i = 1, 2, \ldots, M_b$ *minimize the following set of expressions for a given* λ,

$$d_i(q_i) + \lambda \cdot r_i(q_i), \quad i = 1, 2, \ldots, M_b, \tag{2.18}$$

then it is also a solution of Formulation 2.1 for a given B *equal to*

$$B'(\lambda) = \sum_{i=1}^{M_b} r_i(q_i^*). \tag{2.19}$$

Theorem 2.1 is valid in both continuous space (where q_i takes a continuous range of values) and discrete space (where q_i only defined in some discrete values, as in the MPEG case, where the q_i's are integers between 1 and 31). A proof of this theorem can be found in Chapter 14 of [36] for the continuous case, and in [15] for the discrete case. To find a solution of Formulation 2.1, we search for a λ in (2.18), such that $B'(\lambda)$ meets the specified value of B. The search can be done by the following bisection method.

Algorithm 2.1 *Lagrange Multiplier Method with Bisection Search*

Step 1. *Make an initial guess on* λ_1 *and* λ_2, *with* $\lambda_1 < \lambda_2$.

Step 2. *Substitute* λ_1 *into (2.18) and minimize the expression to derive* q_i^*, $i = 1, 2, \ldots, M_b$. *Substitute* q_i^* *into (2.19) to get* $B'(\lambda_1)$.

Step 3. *Follow the same procedure as in Step 2 for* λ_2 *to get* $B'(\lambda_2)$.

Step 4. *If* $[B'(\lambda_1) - B] \cdot [B'(\lambda_2) - B] > 0$, *i.e, the solution does not fall in between the two initial guess values, go to Step 1 and make another guess. Otherwise, continue to the next step.*

Step 5. *Let* $\lambda_m = (\lambda_1 + \lambda_2)/2$.

Step 6. *Follow the same procedure as in Step 2 for* λ_m *to get* q_i^*, $i = 1, 2, \ldots, M_b$ *and* $B'(\lambda_m)$.

Step 7. *If* $[B'(\lambda_1) - B] \cdot [B'(\lambda_m) - B] < 0$, *substitute* λ_2 *by* λ_m, *otherwise, substitute* λ_1 *by* λ_m.

Step 8. *Check if*

$$\left| \frac{B'(\lambda_m) - B}{B} \right| < \epsilon,$$

where ϵ is a preset small number. If it is true, the optimization is done and the solution is q_i^*, $i = 1, 2, \ldots, M_b$. Otherwise, go to Step 5 for another iteration.

In addition to bisection search, other iterative algorithms for solving nonlinear equations (see Chapter 6 of [12]) can also be used for searching the solution. For the case where q_i's are discrete and finite, a fast search algorithm has been proposed in [50]. Note that in the Lagrange Multiplier Method, all input data have to be collected (for calculating r-q and d-q functions) before any real encoding can take place. A method which used sliding-window to shorten the delay based on *constant slope optimization algorithm* has been proposed in [43].

The computational complexity of Algorithm 2.1 is relatively low because of the assumption of independent quantization, which allows both the R-D measurement and the minimization of (2.18) to be done independently for each block. However, this assumption is not met by an MPEG encoding scheme where rate-distortion values for a given frame depend on previously quantized frames. Therefore, the algorithm can not be applied to MPEG directly. In Chapter 4 and Chapter 5, we still find this algorithm useful in our fast algorithm after the coding dependency is removed.

Note that the Lagrange multiplier λ can also be used as a control parameter instead of q in the buffer state feedback control scheme [8].

2.4.2 Trellis-Based Techniques

Now we consider constraints from the finite buffer size, and still assume that the blocks are independently coded. The same notation as in the previous section is

used. The buffer occupancy is still calculated by (2.1), but considering the fact that the buffer occupancy after block i is coded depends on all the quantization settings prior and up to the block i, a more accurate expression would be

$$b(i, q_1, q_2, \ldots, q_i) = \max\left(b(i-1, q_1, q_2, \ldots, q_{i-1}) + r_i(q_i) - r_b, 0\right), \tag{2.20}$$

with a given initial buffer occupancy $b(0)$. Again, as in (2.1), the max function is used to represent the buffer underflow case, where padding bits are appended. Because the value of buffer occupancy after block i is coded also depends on q's for the previous blocks, it is no longer possible to optimize each block independently.

Formulation 2.2 *Independent Block Coding with Buffer Constraints:*
Determine q_i, $i = 1, 2, \ldots, M_b$ *to*

$$\textit{minimize} \sum_{i=1}^{M_b} d_i(q_i), \tag{2.21}$$

$$\textit{subject to} \quad b(i, q_1, q_2, \ldots, q_i) < b_{max}, \quad i = 1, 2, \ldots, M_b, \tag{2.22}$$

where b_{max} *is the buffer size.*

There are multiple constraints in this formulation. It is still possible to solve the problem with multi-dimensional Lagrange multiplier [6], but it would become too complex in both the formulation and the calculation at a higher dimension. A method based on a forward dynamic programming approach known as Viterbi algorithm has been proposed in [43] to solve this problem. The first step in the algorithm is to build a special trellis graph. The trellis consists of several stages, with each stage corresponding to a block to be coded. A node in a stage is defined as a particular buffer occupancy after the block corresponding to that

stage is coded. With these definitions, the branches, which connect nodes from stage i to stage $i+1$, are grown for every node in stage i and for every possible quantization setting for block i. The cost for each branch is then defined as the distortion produced with the corresponding quantization setting. The problem becomes to find a path in the trellis graph that has smallest total cost, which can be solved by dynamic programming techniques. The buffer constraints are satisfied by "pruning" all the branches which violate the constraints during the growth of the trellis. By this algorithm, the true global optimum solution can be found. However, the number of nodes could grow rapidly with the number of stages, which contributes the increase of both the computational complexity and the memory requirement. Several techniques have also been proposed to reduce the complexity, including the use of buffer state clustering, sliding window, and a fast approximation using a Lagrange multiplier method. These techniques result in a sub-optimum solution which is very close to the global one [43].

2.4.3 Techniques for Dependent Quantization

In the previous two formulations, we assumed that the blocks are coded independently, which is not true for most video compression schemes, including MPEG, where predictive coding is used. In the following formulation, we assume the coding unit is a frame, and prediction between consecutive frames is used for coding (e.g. a IPPPP... structure in MPEG). Suppose there are N frames, and quantization scales q_i, $i = 1, 2, \ldots, N$ are assigned to each frame, respectively. Because of dependency, the number of bits generated by frame i becomes $r_i(q_1, q_2, \ldots, q_i)$, and distortion becomes $d_i(q_1, q_2, \ldots, q_i)$. In the following formulation, we ignore the buffer constraints. Suppose the channel rate is R bits per second and the frame

rate is F frames per second, the total bits available for N frames are $B = N \cdot R/F$.

Formulation 2.3 *Dependent Frame Coding without Buffer Constraints: Determine* q_i, $i = 1, 2, \ldots, N$ *to*

$$\textit{minimize} \sum_{i=1}^{N} d_i(q_1, q_2, \ldots, q_i), \quad \textit{subject to} \quad \sum_{i=1}^{N} r_i(q_1, q_2, \ldots, q_i) \leq B. \tag{2.23}$$

A theorem similar to Theorem 2.1 also holds, except that the cost function for each frame can no longer be minimized independently. To solve the problem with Lagrange multiplier method, we define a set of Lagrangian cost for a given λ as

$$J_i(q_1, q_2, \ldots, q_i) = d_i(q_1, q_2, \ldots, q_i) + \lambda \cdot r_i(q_1, q_2, \ldots, q_i), \quad i = 1, 2, \ldots, N. \tag{2.24}$$

Then, the expression corresponding to (2.18) becomes

$$\min_{q_1, q_2, \ldots, q_N} \sum_{i=1}^{N} J_i(q_1, q_2, \ldots, q_i), \tag{2.25}$$

which has to be minimized in N-dimensional space. We can see that the complexity grows exponentially, not only for the operations in minimization, but also for the evaluation of the rate-quantization and distortion quantization functions, which has to be done over all possible quantization settings.

Several methods have been proposed to solve this problem [46, 6]. The algorithm in [46] is still based on the Lagrange multiplier technique, and inside the procedure, it calls for a trellis-based dynamic programming procedure to solve the multi-dimensional unconstrained minimization problem defined in (2.25). Note that in this algorithm, the definition of trellis nodes is the quantization setting of a frame (while the one used in Section 2.4.2 is the buffer state). In MPEG, for a

general frame structure such as IBBPBBPBB..., and a given set of quantization settings for all I and P frames, the coding of B frames becomes independent and can be independently minimized. Hence, it is not necessary to include B frames in the trellis graph, thus reducing the number of nodes and branches. The complexity can be further reduced with the *Monotonicity* assumption, which is defined as follows [46].

Definition 2.1 *Monotonicity Property*

For a dependent coding system, for any given $\lambda \geq 0$, *if "the quantization step-size of* q_1 *is finer than that of* q_1'*" implies*

$$J_2(q_1, q_2) < J_2(q_1', q_2), \tag{2.26}$$

then the coding system is defined to possess monotonicity property.

This monotonicity property implies that a better quality (finer quantization) in the reference frame will lead to more efficient coding in the rate-distortion sense. Most of the MPEG encoding results in [46] and in our experiments confirm this property. By applying this property, many branches and nodes in trellis can be eliminated, thus saving computations, including the costly evaluation of rate-quantization and distortion-quantization functions associated with these nodes and branches.

Note that buffer constraints are ignored in the algorithm. It is possible to take the buffer constraints into account by checking and "pruning" all the paths which lead to buffer overflow. Also note that the algorithm is complex in that the trellis has to be built and processed for every unconstrained minimization of (2.25), which has to be done for every λ's in the search procedure of Lagrange multiplier

method. Another problem is the number of evaluations on the rate-quantization and distortion-quantization functions, which is in the order of $N_B \cdot (N_P + N_I)^M$, where N_B, N_P and N_I are the number of B, P, I frames in a GOP respectively, and M is the number of choices of quantizers for a frame ($M = 31$ in MPEG). Because these R-D evaluations require to run actual encodings for all quantization settings, they consume a substantial amount of computations, even after "pruning" many of them with the monotonicity assumption. In Chapter 3, we will introduce the use of penalty functions to take the buffer constraints into account, and the use of gradient search to greatly reduce the number of rate and distortion evaluations.

It is also possible to reduce the complexity by approximating the rate and distortion functions without actual encoding process taking place on every quantization settings. A survey of previously proposed approximation models is presented in the next section.

2.5 Approximation Models for Rate and Distortion Functions

As we mentioned in the previous section, although the methods based on the rate-distortion (R-D) optimal techniques usually can achieve better quality, they require a significant amount of computations for measuring the rate and distortion functions, which makes these algorithms impractical in many applications, although they may still be useful for benchmarking purposes. The rate and distortion models provide a way to realize these algorithms, by avoiding the need for measuring the rate-distortion data on all possible quantization settings. Previous

work on rate and distortion modeling has been based to a large extent on the exponential statistics models. In this section, we review and examine the feasibility of applying these models to the optimum bit-rate control problem.

In the following survey, to evaluate the accuracy of the approximation models we use the MPEG-2 encoder of [41] to obtain MSE and code length, for all the quantization settings (from 1 to 31). Both I, P, and B frames are included in the test. For P and B frames, their reference frames are fixed at quantization scale 10. Then we calculate the approximated data from the model. The relative error is then calculated by

$$\text{RelativeError} = \left| \frac{\text{EstimatedValue} - \text{OriginalValue}}{\text{OriginalValue}} \right|. \tag{2.27}$$

2.5.1 Statistical Model for Gaussian Source

Many statistical models are based on the Gaussian source assumption, because it usually leads to simpler expressions.

Theorem 2.2 *Rate-Distortion Function for Gaussian Source*

The rate-distortion function for a zero-mean Gaussian source with variance σ^2 is

$$r(d) = \begin{cases} (1/2) \cdot \log(\sigma^2/d), & 0 \leq d \leq \sigma^2, \\ 0, & d > \sigma^2, \end{cases} \tag{2.28}$$

where d is distortion measured by squared error.

A proof of Theorem 2.2 can be found in [11]. Note that the rate $r(d)$ in (2.28) is defined in bits-per-pixel. If we still assume that the linear relationship

of distortion-quantization as in Section 2.3.1 holds, i.e.,

$$d(q) = m \cdot q, \tag{2.29}$$

substituting (2.29) into (2.28) and multiplying by the total number of pixels $W \cdot H$, we get

$$r(q) = \alpha + \beta \cdot \log \frac{1}{q}, \tag{2.30}$$

with

$$\alpha = W \cdot H \cdot \frac{1}{2} \cdot \log \frac{\sigma^2}{m}, \tag{2.31}$$

$$\beta = W \cdot H \cdot \frac{1}{2}. \tag{2.32}$$

This model has been used in many contexts where a rate-distortion function is required [14]. To estimate the model parameters, the variance σ^2 can be measured directly from the source data, and the value of m has to be estimated in some way. The problems of this model when applying it to MPEG coding are that a general DCT transformed video data is usually not a Gaussian source, and the linear assumption in (2.29) does not generally hold (see Section 2.3.3), especially for P and B frames.

To evaluate "how good" this model is for MPEG coding, we select several frames (with different type) from several different video sequences, and measure their rate-quantization functions by encoding the frames at all 31 quantization settings. For each frame, the measured data is then used to derive the "best possible" values of α and β for the frame, using curve fitting which minimizes the error between the curve and actual data. This represents the best achievable

accuracy of the model for a given frame. A test case which uses a P frame from Miss America sequence is shown in Fig. 2.9, where both the actual data and the best-fitted curves are plotted. For the approximated curve, the value of parameters are $\alpha = 1.54 \times 10^5$ and $\beta = 5.21 \times 10^4$. Note that the value of β is different from the one in (2.32), which is $352 \cdot 240/2 = 4.22 \times 10^4$.

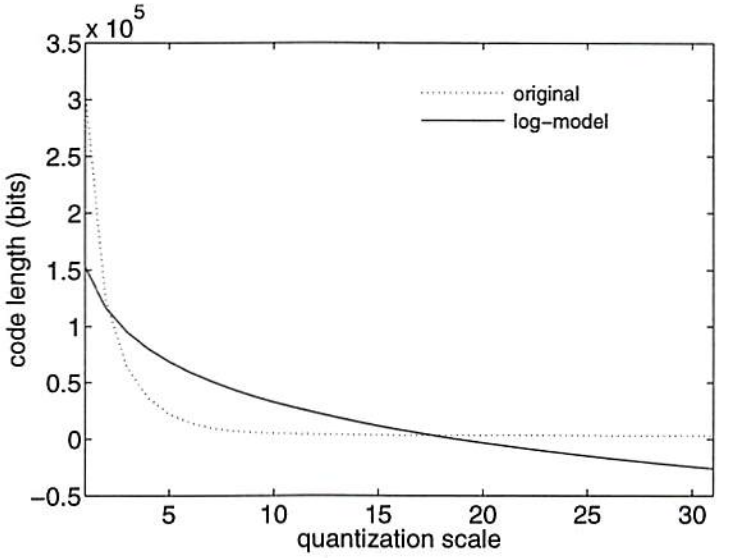


Figure 2.9: The best possible approximation with logarithm model for a Miss America P-frame, derived by curve fitting. $\alpha = 1.54 \times 10^5$, $\beta = 5.21 \times 10^4$.

Some maximum and average relative errors are calculated by (2.27). The results show that the average relative error is about 40% for a typical I frame and 150% to 400% for P and B frames, and the maximum relative errors are several times larger. A set of data calculated over all frames and different frame types are shown in Table 2.1, under the column *opt.log*. We observe that the relative errors from this model are too large to be useful.

2.5.2 Exponential Model

To cope with the difficulties of the model of (2.30), a new model has been proposed specifically for MPEG coding in [13, 14]. In the model, the approximation function is defined as

$$r(q) = \alpha + \frac{\beta}{q^{\gamma}}. \tag{2.33}$$

The performance is better because of a third parameter γ, which controls the curvature of the function, is added.

We also evaluate the best achievable performance for this model, in the same way as we did in Section 2.5.1, i.e., we use curve fitting to find "optimum parameters" for each given frame. A test case which also uses the same P frame from Miss America sequence is shown in Fig. 2.10. For the approximated curve in the figure, the values of parameters are $\alpha = 2.09 \times 10^3$, $\beta = 3.11 \times 10^5$, and $\gamma = 1.456$.

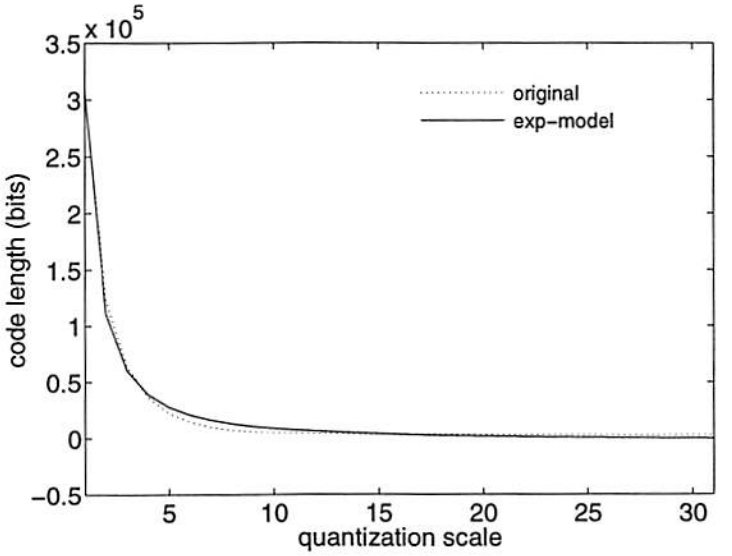


Figure 2.10: The best possible approximation with exponential model for a Miss America P-frame, derived by curve fitting. $\alpha = 2.09 \times 10^3$, $\beta = 3.11 \times 10^5$, $\gamma = 1.456$.

The average and maximum relative errors from several different frames are

also shown in Table 2.1, in the column labeled as "opt.expon" (for "optimum exponential"). The results are better but the errors are still somewhat large. The larger error usually comes from the B and P frames of low activity video sequence, because the function cannot model the characteristic of prediction residues well.

	Intra-frames Relative Errors for BITS(q)			
	average error		maximum error	
	opt.log	opt.expon	opt.log	opt.expon
Claire	133.43%	5.77%	615 73%	28 46%
Football	107.72%	14.95%	538.76%	77.09%
Miss America	207.01%	26.75%	934.64%	100.03%
Susie	208.04%	21.15%	749.56%	68.00%

Table 2.1: Relative errors for intra-frame approximation functions using exponential models. The statistic is over the entire quantization scale range, and over three type of frames (I, P, B). opt.log: optimum logarithm, opt.expon: optimum exponential.

Several problems arose when we attempted to put this model into practical implementation. First, the model is derived from the assumption that the distortion (measured by MSE) is linearly proportional to q, which will contribute to large errors for the distortion-quantization functions. Second, it is not possible to derive the "optimum" parameters without knowing the actual R-D data first, so the parameters are either fixed at some empirical value or adaptively adjusted based on some statistics measured from the source. Hence, the actual results could be much worse than the optimum ones derived above. Although the model might be useful to determine the bit-allocation of frames in a predictive feedback control algorithm [14], the model error is still too large to be useful for optimum rate control algorithms presented in the previous section.

2.5.3 Other Models

Several other statistical models have also been proposed or used in their specific rate control algorithm. For example, in [18, 5, 25], exponential expressions other than (2.33) were used to model the relationship between the rate, distortion, and quantization step size in a macroblock. Many of them do not take into account the dependencies that arise in the choice of quantizers for the reference frames and the predicted frames. Even when these models take the dependencies into account, as in [53], they ignore some non-linear effects that are typical in video coding. For example, under the general intra/inter selection rule, there is no dependency if the quality of the reference frame is too low, because blocks are coded in intra mode if the prediction is not sufficiently good. Therefore, in general, previous models are not suitable for the R-D optimization based rate-control algorithm.

In Chapter 4, we will introduce an approximation method using interpolation functions and taking into account the inter-frame dependencies.

2.6 Conclusions

In this Chapter, we first surveyed rate control schemes based on direct buffer state feedback. Then, we reviewed and analyzed the rate control algorithm defined in MPEG Test Model, known as TM5, and showed that their assumptions on R-D characteristics do not usually hold. Although the algorithm with default setting of parameters can get good encoding quality for some video sequence, it can not be applied for any given video sequence and any given rate. It is also difficult to adjust parameters to improve the quality without many test encoding. Another problem is, it does not control the buffer to prevent the encoder buffer from

overflow.

We next reviewed another class of bit-rate control algorithms which use rate-distortion optimization, including Lagrange multiplier technique and trellis based technique, and an algorithm which combines both and takes dependency coding into account. With the algorithm, it is possible to derive the optimum solution for any given video sequences and any given channel conditions. However, the costly computation complexity, mainly from the evaluation of rate and distortion functions for many quantization settings, makes the algorithm impractical for many applications, although it is a good benchmark reference for other rate control techniques.

Finally we surveyed approximation models for rate and distortion, which, if feasible, can save a lot of computations for the optimum rate control algorithm. We tested both Gaussian and exponential models using real video frames from several different sequence. The results showed that the relative model errors are too large to be useful for the optimum rate control algorithm.

In the follow-up chapters, we will present (i) our approach of optimum rate control which makes practical implementation possible, (ii) our techniques to the rate-distortion approximation which can be applied to optimum rate control algorithms, and (iii) a fast method which makes it possible to achieve the quality of optimum rate control within near real-time encoding environments.

Chapter 3

Bit-Rate Control Using Gradient Search

3.1 Introduction

In this dissertation, our first objective is to develop a practical method to achieve a good and robust solution to the rate control problem. We follow the delayed-decision rate-distortion optimization approach as introduced in Section 2.4, and formulate the bit-rate control as a constrained optimization problem. To solve the problem, the dynamic programming technique presented in Section 2.4.3 can be used. Although, in theory, the global optimum solution can be achieved, the algorithm needs the rate-distortion (R-D) data for many possible quantization settings. To measure those data, the input video has to be repeatedly encoded for each quantization setting, which requires a significant amount of computations. For example, for a small GOP of just 6 frames, the amount of computations is equivalent to that of encoding about 6×10^4 frames, and the number grows

exponentially as GOP size increases. In the current context, we do not intend to obtain the global optimal solution. Instead, we only look for reasonable sub-optimal solutions while reducing the computation cost to a practical level. In this chapter, we present a method which is based on using penalty functions and an iterative gradient search and only requires the R-D data to be computed along the search path [32, 33].

This chapter is organized as follows. In Section 3.2 we first formulate the buffer control as a nonlinear constrained optimization problem. In Section 3.3 we approximate the solution by using penalty functions and a gradient search method. In Section 3.4 we present some results, where a short GOP sequence is used to demonstrate that the performance is close to the optimal point achievable by exhaustive search. We also apply our technique to encode short image sequences and compare our performance with that achieved by the TM5 algorithm. Note that although we provide experimental results for an MPEG encoder, the algorithms presented here are applicable with simple modifications to other video encoding scenarios as well. Also note that this algorithm still takes considerable computations. Techniques to further reduce the complexity are presented in the next chapter.

3.2 Problem Formulation

We use a Group of Pictures (GOP) as our basic rate control unit. In addition to the usual buffering constraints, we also impose a constraint to maintain a strictly constant number of bits for each GOP. This constraint is required for recording compressed stream on a digital tape recorder [58], and also allows faster searching

and indexing for video streams stored in a CD-ROM or hard drive. The goal of our approach is to maintain constant bit rate for every GOP, keep the buffer occupancy within constraints, while minimizing the overall distortion.

As mentioned in the previous chapter, rate control can be achieved by a Global Control procedure, which assigns q for each frame, followed by an Adaptive Quantization procedure, which modulates the q to determine *mquant* for each macroblock. Although we only consider the Global Control procedure in this section, our method applies to choosing one among a finite set of operating points. Thus it would be easy to extend our algorithm to the case where the operating points are defined by their various "quality levels", with adaptive q allocations performed for each frame and quality level, as in [42]. For now, we only use constant quantization for each frame, to demonstrate the performance of our technique. Under this assumption, the buffer control problem is to assign the quantization scale q_i for the i^{th} frame in a GOP, as shown in Fig. 3.1, such that the overall quality, measured by a pre-defined cost function, is optimized.

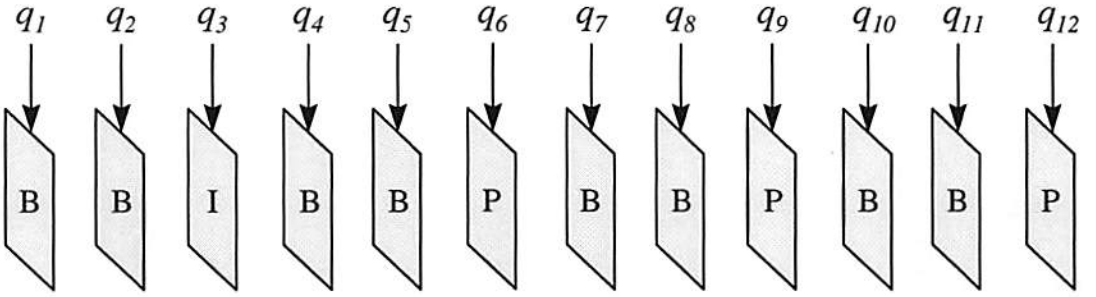


Figure 3.1: Quantization assignment for global control

Because MPEG is a dependent coding system, we use vector notation for the set of quantization scales in a GOP, denoted as $\mathbf{q} = (q_1, q_2, \ldots, q_N)^T$. By

using a vector expression for $\mathbf{q}$, we are taking into account the "dependency" of the problem, i.e., the distortion/rate trade off for predicted/interpolated frames depends on the frames that were used to generate the prediction [46]. In the formulation, the following notation is used:

R : channel rate in bits per second.

F : frame rate in frame per second.

N : number of frames in a GOP.

i : index to a frame.

$\mathbf{q}$: quantization setting assigned to a GOP, i.e.,
$\mathbf{q} = (q_1, q_2, \ldots, q_N)^T$, where q_i represents the quantization scale assigned to frame i.

$r_i(\mathbf{q})$: number of bits generated by frame i, when the value of $\mathbf{q}$ is used to encode the GOP.

$d_i(\mathbf{q})$: distortion introduced by quantization of frame i, when the value of $\mathbf{q}$ is used to encode the GOP. Again, any kind of meaningful distortion measure can be used here, including the MSE defined in (2.4), which is the one used in our experiments.

$b(i, \mathbf{q})$: buffer occupancy after the i^{th} frame is coded, when the value of $\mathbf{q}$ is used to encode the GOP.

The buffer occupancy after frame i is coded can be calculated by

$$b(i, \mathbf{q}) = \max\left(b(i-1, \mathbf{q}) + r_i(\mathbf{q}) - \frac{R}{F}, 0\right), \quad \forall i = 1, 2, \ldots, N. \tag{3.1}$$

With the requirement of strictly constant number of bits for each GOP, the initial buffer occupancy should be $b(0, \mathbf{q}) = 0$ for all GOP's. Note that the max function in the formulation is needed to reflect underflow situations, where the buffer occupancy never falls below zero since stuffing bits are appended to the output stream to prevent underflow. To maintain a constant number of bits per GOP, the final buffer occupancy $b(N, \mathbf{q})$ should be zero (possibly after adding stuffing bits).

We define the cost function as

$$J(\mathbf{q}) = D(\mathbf{q}) + wE(\mathbf{q}), \tag{3.2}$$

where $D(\mathbf{q})$ represents the average distortion in the GOP and is defined as

$$D(\mathbf{q}) = \frac{1}{N}\sum_{i=1}^{N} d_i(\mathbf{q}), \tag{3.3}$$

$E(\mathbf{q})$ is the average squared difference in distortion between consecutive frames,

$$E(\mathbf{q}) = \frac{1}{N}\sum_{i=1}^{N}[d_i(\mathbf{q}) - d_{i-1}(\mathbf{q})]^2, \tag{3.4}$$

and w is the weighting coefficient between $D(\mathbf{q})$ and $E(\mathbf{q})$. The purpose of $E(\mathbf{q})$ in the cost function is to minimize the abrupt changes in quality and avoid the "flicker problems". When $w = 0$, the cost function is simply an average distortion similar to the one used in Section 2.4.

The rate control problem is formulated as below.

Formulation 3.1 *Dependent Block Coding with Buffer Constraints:*

Determine $\mathbf{q}^* = (q_1^*, q_2^*, \ldots, q_N^*)^T$ *such that*

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} J(\mathbf{q}), \tag{3.5}$$

subject to

$$q_i \in \{1, 2, \ldots, 31\}, \quad i = 1, 2, \ldots, N, \tag{3.6}$$

$$b(i, \mathbf{q}) \leq b_{max}, \quad i = 1, 2, \ldots N-1, \tag{3.7}$$

$$b(N, \mathbf{q}) = 0, \tag{3.8}$$

where b_{max} *is the prescribed maximum buffer size.*

Note that the constraint in (3.8) is required to maintain a constant number of bits per GOP.

3.3 Solution Using Gradient Search Techniques

The optimization problem formulated in the previous section is an integer programming problem with a nonlinear cost function and nonlinear constraints. These characteristics make the problem difficult to solve efficiently. In order to reduce the complexity, our first approximation is to change the integer-valued variable into a continuous one, so that many optimization techniques defined in continuous domain [36, 16] can be applied.

3.3.1 Penalty Functions

The constraints of (3.7) and (3.8) can be taken into account by adding penalty functions to the cost, $J(\mathbf{q})$. The penalty functions are defined as

$$P_i(\mathbf{q}) = \{\max[(b(i,\mathbf{q}) - b_{max}), 0]\}^2, \tag{3.9}$$

$$Q(\mathbf{q}) = b(N,\mathbf{q})^2, \tag{3.10}$$

where $P_i(\mathbf{q})$ penalizes exceeding the maximum buffer size and $Q(\mathbf{q})$ favors a nearly empty buffer at the end of GOP. The new cost function is then defined as

$$\phi(\mathbf{q}, c) = J(\mathbf{q}) + c\left(\sum_{i=1}^{N-1} P_i(\mathbf{q}) + Q(\mathbf{q})\right) \tag{3.11}$$

where the parameter c determines the amount of the penalty. The original problem can be approximated by iteratively solving the unconstrained problem of minimizing $\phi(\mathbf{q}, c)$ as $c \to \infty$. This can be done by either assigning a single large value to c and solving the optimization problem *(Single-Pass Penalty)*, or using iteratively increasing c as shown in the following procedure *(Iterative Penalty)*:

Algorithm 3.1 *Iterative Penalty*

Step 1. *Initialize* $k = 1$, *and preset* $c_k = c_1$ *to a small value.*

Step 2. *Minimize* $\phi(\mathbf{q}, c_k)$ *with variable* $\mathbf{q}$. *Denote the solution as* $\mathbf{q}^*$.

Step 3. *Check if the constraints are violated, i.e.,*

$$\sum_{i=1}^{N-1} P_i(\mathbf{q}^*) + Q(\mathbf{q}^*) < \epsilon$$

where ϵ is a preset small number. If the inequality is met, the optimization is done and the solution is $\mathbf{q}^*$. Otherwise, assign $c_{k+1} = g \cdot c_k$ (g is also a preset factor, usually 10), increase k by one, and go to Step 2.

The convergence of Algorithm 3.1 is based on the following theorem.

Theorem 3.1 *Penalty Function Convergence*
In the Algorithm 3.1, if c_k increases in each iteration, then

1. *$\phi(\mathbf{q}^*, c_k)$ is non-decreasing,*
2. *$\sum_{i=1}^{N-1} P_i(\mathbf{q}^*) + Q(\mathbf{q}^*)$ is non-increasing,*
3. *$J(\mathbf{q}^*)$ is non-decreasing,*

Also, if $c_k \longrightarrow \infty$, then $\sum_{i=1}^{N-1} P_i(\mathbf{q}^*) + Q(\mathbf{q}^*) \longrightarrow 0$, and $\mathbf{q}^*$ converges to the solution of Formulation 3.1.

A proof of Theorem 3.1 can be found in Chapter 12 of [16].

3.3.2 Iterative Gradient Search

In order to solve the unconstrained problem efficiently, we make an assumption that the cost function is smooth. Because the encoding process is a nonlinear operation, it is not easy to justify the smoothness of the cost function. To visualize the shape of the function, we encode two frames of Miss America sequence with every possible setting of quantization scales, and calculate the cost function. The surface plot of the function is shown in Figure 3.2. From the figure, we observe a smooth surface for the cost function and thus can resort to iterative gradient methods. In a real situation where several frames are involved, the function may

not be perfectly smooth, but we still can reach a reasonably good sub-optimal point by the iterations.

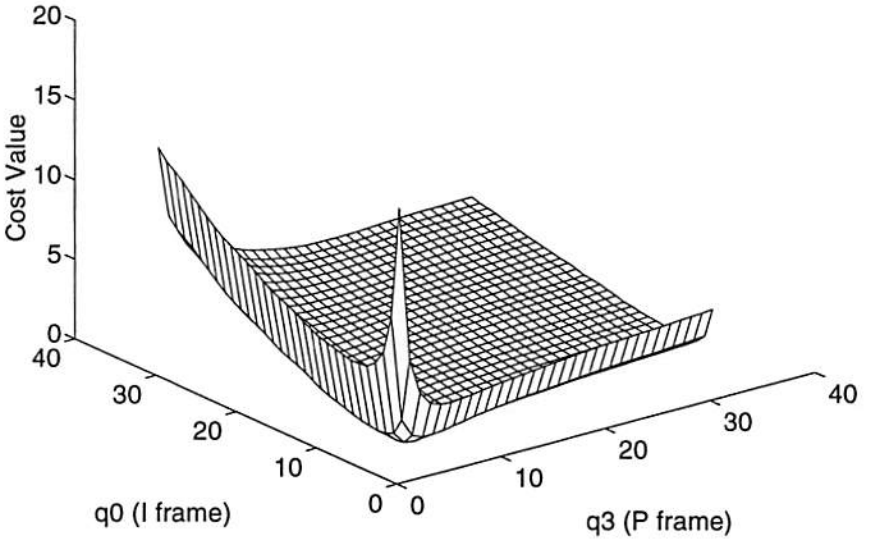


Figure 3.2: Typical cost function in two dimensional case

There are several iterative gradient search algorithms available for our problem [36, 16]. The first one that we have considered is *cyclic coordinate descent method*, also known as *alternating variables method*. This method minimizes the cost function with respect to one coordinate component q_i at a time. There are two strategies for searching the solution. The first search strategy is to search for a minimum for each coordinate component, as shown in Algorithm 3.2 *(CCD-LS)* (see Fig. 3.3(a) for a search path in discrete case).

Algorithm 3.2 *CCD-LS: Cyclic Coordinate Descent with Line Search*

Step 1. *Initialize the index variable* $k = 1$. *Also make an initial guess for* $\mathbf{q} = (q_1, q_2, \ldots, q_N)$.

Step 2. *Search for q_k such that the cost $\phi(\mathbf{q})$ in (3.11) is minimized, i.e.,*

$$\min_{q_k} \phi(q_1^*, q_2^*, \ldots, q_{k-1}^*, q_k, \ldots, q_N). \tag{3.12}$$

Denote the solution as q_k^*. Note that $q_1^*, q_2^*, \ldots, q_{k-1}^*$ are the variables that have been optimized in the iterations before k.

Step 3. *Increase k by one, if $k < N$, go to Step 2.*

Step 4. *Now we have completed a round for all variables, so we need to check for convergence. Denote $(q_1^*, q_2^*, \ldots, q_N^*)$ as $\mathbf{q}^*$. Check if*

$$\left| \frac{\phi(\mathbf{q}^*) - \phi(\mathbf{q})}{\phi(\mathbf{q})} \right| < \epsilon$$

where ϵ is a preset small value. If the inequality is met, the optimization is done and the solution is $\mathbf{q}^*$. Otherwise, assign $k = 1$, $\mathbf{q} = \mathbf{q}^*$, and go to Step 2 for the next iteration.

Another search strategy is to only allow to search within a fix step size for each coordinate component in each iteration. The search path with this strategy will be closer to a steepest descent search, and is expected to have better convergence property. The procedure is shown in Algorithm 3.2 *(CCD-US)* (see Fig. 3.3(b) for a search path in discrete case).

Algorithm 3.3 *CCD-US: Cyclic Coordinate Descent with Unit Stepping*

Step 1. *Initialize the index variable $k = 1$. Also make an initial guess for $\mathbf{q} = (q_1, q_2, \ldots, q_N)$.*

Step 2. *Search for q_k such that*

$$\phi(q_1^*, q_2^*, \ldots, q_{k-1}^*, q_k, \ldots, q_N)$$

is minimized, subject to

$$q_k - 1 \leq q_k^* \leq q_k + 1,$$

where the notations are the same as in Algorithm 3.2.

Step 3. *Increase k by one, if $k < N$, go to Step 2.*

Step 4. *Check for convergence: Denote $(q_1^*, q_2^*, \ldots, q_N^*)$ as $\mathbf{q}^*$. Check if*

$$\left| \frac{\phi(\mathbf{q}^*) - \phi(\mathbf{q})}{\phi(\mathbf{q})} \right| < \epsilon$$

where ϵ is a preset small value. If the inequality is met, the optimization is done and the solution is $\mathbf{q}^*$. Otherwise, assign $k = 1$, $\mathbf{q} = \mathbf{q}^*$, and go to Step 2 for the next iteration.

The advantage of this coordinate descent method is its simplicity. However, there is not a strong theoretical background for its convergence, as described in Chapter 7 of [36]. We consider this method because it can avoid the computation for gradient, which is costly in our rate control problem.

The second method that we have applied is the *steepest descent method.* In this method, we search along the direction pointed by the negative gradient vector

$-\nabla\phi(\mathbf{q})^T$, and update $\mathbf{q}$ with

$$\mathbf{q}_{k+1} = \mathbf{q}_k - \alpha_k \nabla\phi(\mathbf{q}_k)^T \tag{3.13}$$

where α_k is a nonnegative scalar value. Similar to cyclic coordinate descent, there are also two strategies for searching the solution, either searching for a minimum along the search direction, as shown in Algorithm 3.4 *(STD-LS)* (see Fig. 3.3(c) for a search path in discrete case), or only allowing to search within a fixed step size along the direction, as shown in Algorithm 3.5 *(STD-US)* (see Fig. 3.3(d) for a search path in discrete case).

Algorithm 3.4 *STD-LS: Steepest Descent with Line Search*

Step 1. *Initialize the index variable* $k = 1$. *Also make an initial guess for* $\mathbf{q}_k = \mathbf{q}_1$.

Step 2. *Approximate the gradient vector* $\nabla\phi(\mathbf{q}_k)$ *using first order difference for each coordinate component.*

Step 3. *Define the function* $\varphi(\alpha) = \phi\left(\mathbf{q}_k - \alpha\nabla\phi(\mathbf{q}_k)^T\right)$, *which represents a parametric line along the negative direction of gradient vector.*

Step 4. *Search for a positive* α, *such that* $\varphi(\alpha)$ *is minimized. The solution is denoted as* α_k.

Step 5. *Calculate* $\mathbf{q}_{k+1} = \mathbf{q}_k - \alpha_k \nabla\phi(\mathbf{q}_k)^T$.

Step 6. *If* $k = 1$, *skip the convergence checking and go to Step 2 directly. Otherwise, check if*

$$\left|\frac{\varphi(\alpha_k) - \varphi(\alpha_{k-1})}{\varphi(\alpha_k)}\right| < \epsilon.$$

If the inequality is met, the optimization is done and the solution is $\mathbf{q}_{k+1}$. *Otherwise, increase* k *by one, then go to Step 2 for the next iteration.*

Algorithm 3.5 *STD-US: Steepest Descent with Unit Stepping*

Step 1. *Initialize the index variable* $k = 1$. *Also make an initial guess for* $\mathbf{q}_k = \mathbf{q}_1$.

Step 2. *Approximate the gradient vector* $\nabla\phi(\mathbf{q}_k)$ *using first order difference for each coordinate component.*

Step 3. *Define the function* $\varphi(\alpha) = \phi\left(\mathbf{q}_k - \alpha\nabla\phi(\mathbf{q}_k)^T\right)$, *which represents a parametric line along the negative direction of gradient vector.*

Step 4. *Search for a positive* α *within a range defined by*

$$\|\alpha\nabla\phi(\mathbf{q}_k)\|_\infty \leq 1$$

such that $\varphi(\alpha)$ *is minimized. The solution is denoted as* α_k.

Step 5. *Calculate* $\mathbf{q}_{k+1} = \mathbf{q}_k - \alpha_k\nabla\phi(\mathbf{q}_k)^T$.

Step 6. *If* $k = 1$, *skip the convergence checking and go to Step 2 directly. Otherwise, check if*

$$\left|\frac{\varphi(\alpha_k) - \varphi(\alpha_{k-1})}{\varphi(\alpha_k)}\right| < \epsilon.$$

If the inequality is met, the optimization is done and the solution is $\mathbf{q}_{k+1}$. *Otherwise, increase* k *by one, then go to Step 2, for the next iteration.*

The steepest descent algorithm guarantees convergence to a local minimum, and the difference of step-size between Algorithm 3.4 and Algorithm 3.5 affects the

convergence rate. A proof of the convergence property can be found in Chapter 6 of [36].

Besides the above two approaches, there are several other methods that aim at improving convergence rate, including the Conjugate Gradient and Quasi-Newton methods. These methods are generally preferred for cost functions that include penalties, as in our formulation, because convergence is usually slow when the penalty parameter c is large (see Chapter 12 of [36]). Unfortunately, those methods require a second derivative matrix, or Hessian, either by a direct computation or by an indirect estimation, which is, in our case, costly to compute. Also, because we do not have a perfectly continuous cost function, the error in the approximation is subject to be amplified during the computation of second derivative, which might cause the algorithm to become unstable. So, we do not use these more advanced algorithms at the current stage, and leave a study of their feasibility for future research.

3.3.3 Integer Approximation

In our formulation, the variable $\mathbf{q}$ is only defined on a discrete integer grid. However, the gradient-based optimization algorithms have to be operated in continuous space. A method to cope with this problem would be to define an interpolation scheme to reconstruct the cost function in continuous space based on the available discrete data. This is what we will do in the next chapter, where an approximated R-D model defined in continuous space is used. In this chapter we do not rely on any interpolation models, and so, the algorithms are modified to make them only search through the available data on the integer grid. The search paths of the four search algorithms are shown in Fig. 3.3 in the two-dimensional case. As

shown in the figure, for the cyclic coordinate method, the search direction is always parallel to one of the coordinate axes and hence all the search points will always fall on the integer grid. For the steepest descent method where the search direction can be arbitrary, the search points no longer fall on the integer grid. In this case, we create a search path consisting of those points that are closest to the line, as shown in Fig. 3.3. Those points can be efficiently derived by a line drawing algorithm used in raster computer graphics [17].

3.3.4 Initialization

One problem of the gradient search is that, it only converges to a local minimum solution. Hence initialization is an important factor for the performance. It also affects the rate of convergence. Before coding the first GOP, because we do not know the characteristics of the input sequence, the first initial guess is based on common knowledge for typical quantization scales. Perhaps the best way is to encode the first GOP by the dynamic programming technique [43] to obtain the true global optimum solution, but here, we just set the initial guess to a fixed set of typical values, for example, 8 for I frame, 10 for P frames, and 14 for B frames. For the succeeding GOPs, we can take advantage of the similarity between adjacent GOPs, and use the solution of the previous GOP for the initial guess. In this case, because the solution is usually close to that of previous GOP, the rate of convergence is faster, and thus the computational complexity is reduced.

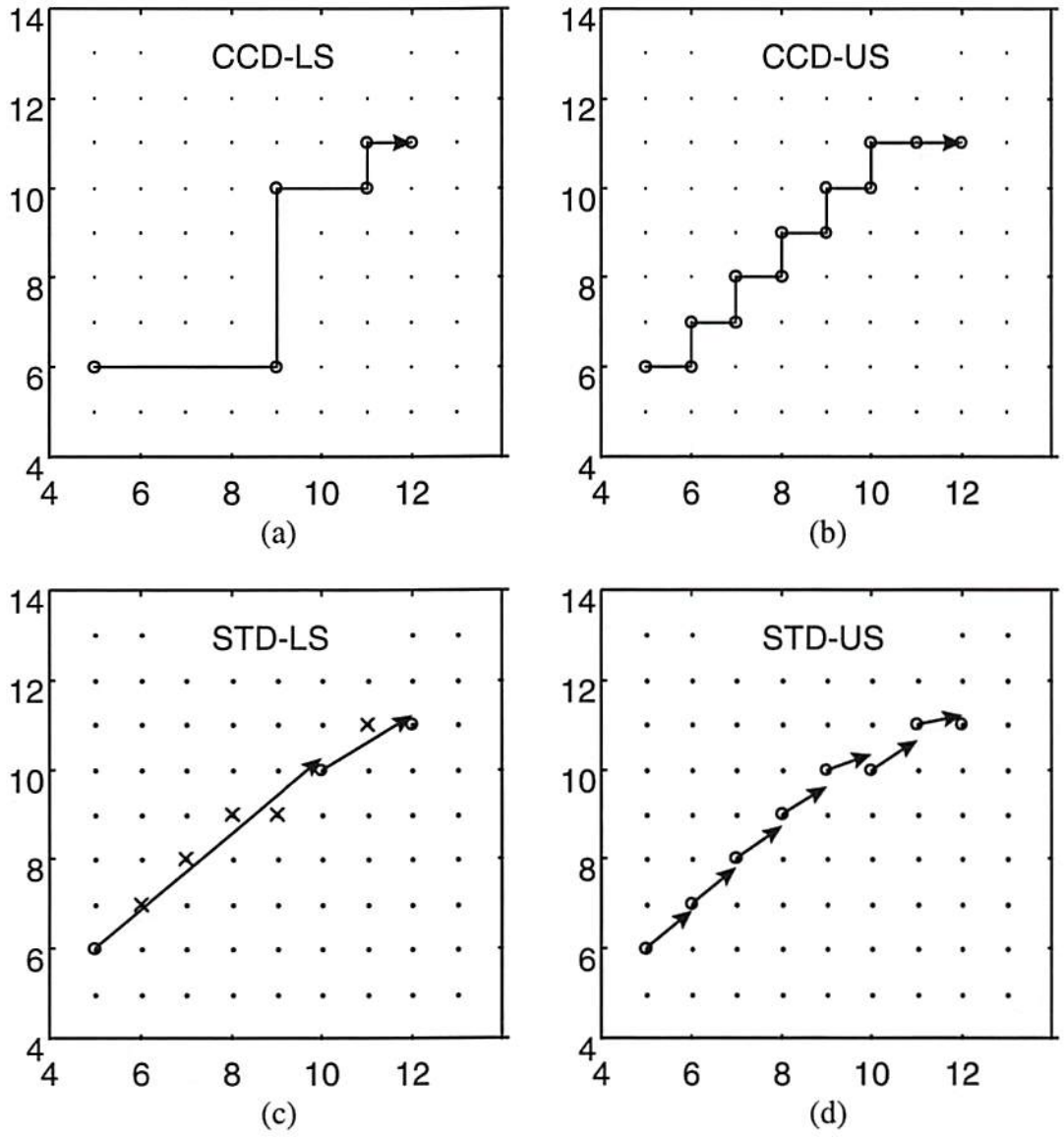


Figure 3.3: Discrete line search paths in 2-D case for the four search algorithms. Figure (a) and (b) show the paths of cyclic coordinate method, where the search points always fall on the integer grid. Figure (c) and (d) show the paths of steepest descent method. In this case, the solid line indicates a path along negative gradient direction. The actual search path is along those points indicated by the circle dot.

3.4 Experiments and Results

Our first experiment is to compare several approaches of the iterative search processes. The objective is twofold: to study their relative performance, and to find out how much they deviate from the optimal solution. We thus consider an experiment on a sub-GOP, which is a set of four frames, with frame type assigned as I, B, B, P respectively. Because at this scale the optimal solution can be derived by exhaustive search, we can compare each of our methods with the optimal solution. We then test the algorithms by encoding short video clips.

3.4.1 Evaluation of Search Algorithms

In this experiment, we choose a set of four frames from Miss America video sequence, and then encode it by an MPEG encoder with all possible combinations of quantization settings to measure the actual distortions and rates in all cases. A modified version of MPEG-1 Encoder version 1.3 from U. C. Berkeley [39] is used for the encoding. To get a reasonable value for the weighting factor w in (3.2) for our first experiment, we calculate and set it to be the ratio between the two standard deviations of $D(\mathbf{q})$ and $E(\mathbf{q})$ over all possible quantization settings.

The first encoding experiment, denoted as *Global*, is to use exhaustive search over all combinations of quantization settings to obtain the global optimal solution. The result serves as a reference for the other methods. Then, the four methods, *CCD-LS*, *CCD-US*, *STD-LS*, and *STD-US*, are tested. For the penalty parameter c in (3.11), the following two choices are used in our experiment: either use a fixed value at $c = 0.05$, or use an iterative strategy as shown in Algorithm 3.1 with an initially setting $c = 10^{-10}$.

The above procedures are tested over several different rate settings and the results are summarized in Figure 3.4 and Table 3.1. The computation complexities are also presented in the table, where the unit of computation is a one-pass encoding (equivalent to encoding 4 frames of video).

	$c = 0.05$		
Method	Relative Error	PSNR (dB)	Complexity
GLOBAL		38.3024	1.5×10^4
CCD-LS	0.04234	38.1580	16.29
CCD-US	0.03671	38.1529	14.08
STD-LS	0.01922	38.2198	15.46
STD-US	0.02851	38.1884	20.13

	iterated on c		
Method	Relative Error	PSNR (dB)	Complexity
GLOBAL		38.3024	1.5×10^4
CCD-LS	0.01062	38.2558	68.46
CCD-US	0.00685	38.2738	50.08
STD-LS	0.00163	38.2893	81.25
STD-US	0.00163	38.2874	117.25

Table 3.1: Four frame experimental result. The *relative error* is the relative difference of the cost value between the specific search method and the global solution. The *complexity* is the total computation requirement relative to one-pass encoding (where 4 frames are coded). All the values are the average over the results from the six test cases, at bit rates 0.8333, 0.9167, 1.0, 1.0833, 1.1667, and 1.25 Mbps, respectively.

Based on the results, we have the following conclusions.

- A good approximation to the optimal solution can be obtained at a fraction of the complexity. Compared to the exhaustive search, the reduction in complexity is of the order of 10^3.

- The method that iterates over the penalty parameter c has much better approximation to the optimal solution, at the expense of about 5 times

higher computational complexity over the method that uses only one fixed c.

- Among these search methods, the steepest descent with line search strategy has the best performance.

- The computation complexity seems to be at least 15 times higher than in the single pass encoding algorithm. In actual implementation, it can be lower than this value. For example, the costly motion estimations can be done by using the original frames and only have to be done once during the whole encoding process. Furthermore, we can take advantage of the fact that there is no frame dependency on the quantization setting of the B frames when all I and P frames are fixed.

By a trade off between the performance and computation complexity, in the following video encoding simulations, we choose the steepest descent with line search strategy, and use one fixed penalty parameter at $c = 0.05$ in the cost function.

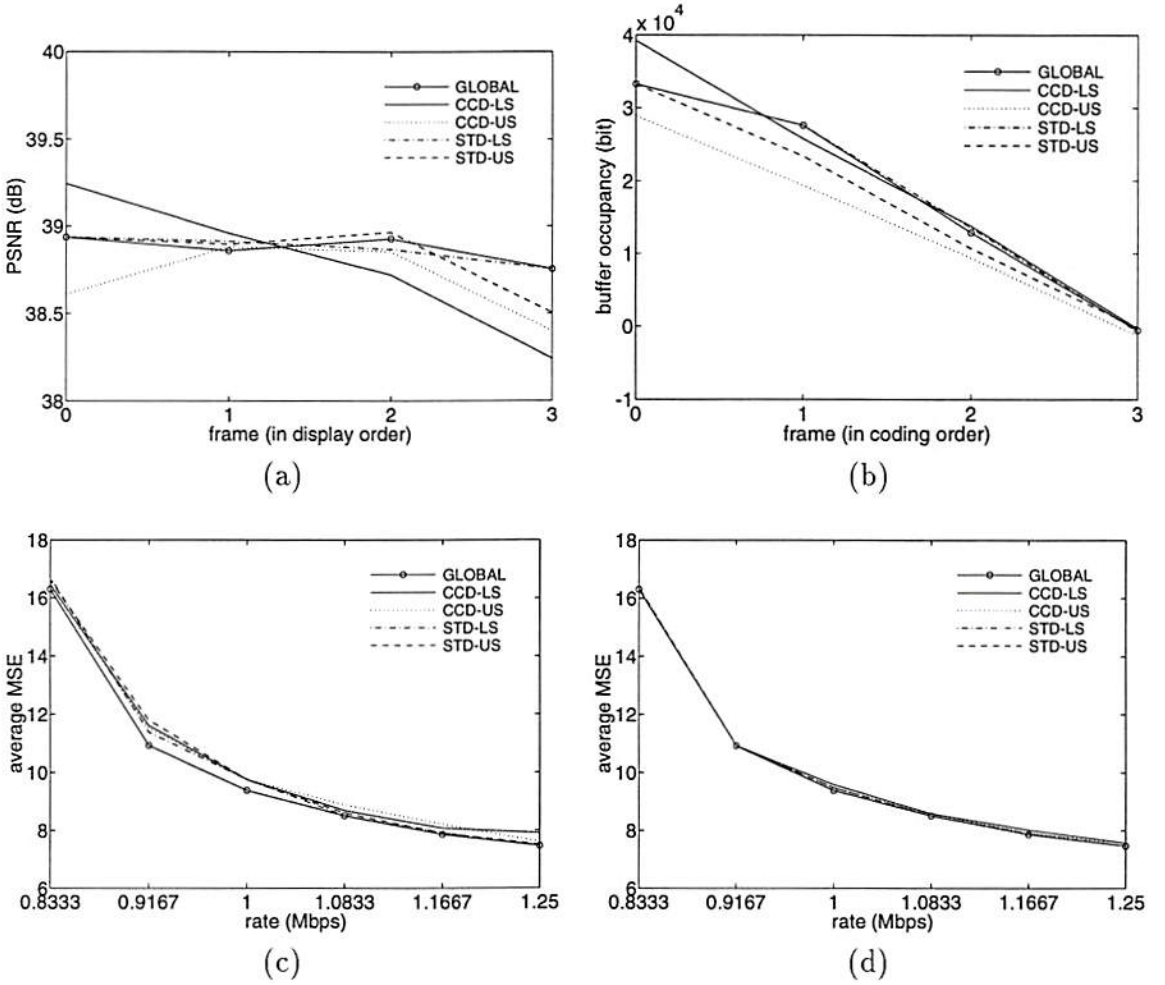


Figure 3.4: *Four frames experiment* (a) PSNR at bit rate 1.0833 Mbps, (b) buffer occupancy at bit rate 1.0833 Mbps, (c) rate-distortion curve. (a), (b), (c) are the results for single-pass penalty method, where c is fixed to 0.05. (d) rate-distortion curve for iterative penalty method, where c is iterated from 10^{-10}, multiplied by 10 after each iteration, until the solution converges and the constraints are satisfied. In each figure, *GLOBAL*: exhaustive search to obtain the global optimum solution; *CCD-LS*: cyclic coordinate descent with line search strategy; *CCD-US*: cyclic coordinate descent with unit stepping strategy; *STD-LS*: steepest descent with line search strategy, and *STD-US*: steepest descent with unit stepping strategy.

3.4.2 MPEG Encoding

In this part of the experiment, two video clips in CIF format are used: the *Football* sequence, 352×240 pixels, 30 frames per second, and the *Miss America* sequence, 352×288 pixels, 25 frames per second. For each video clip, 22 frames are used for the encoding. To do the experiment, we use a program derived from MPEG-2 encoder version 1.1a, published by MPEG Software Simulation Group [41]. One major modification is that the motion estimation is done by using the original frames and is independent of the quantization setting. By doing so, we avoid the motion search on every iteration in the steepest descent method and speed up the experiment. Although the modification will degrade the performance, the degradation is less then 0.1 dB in all our test cases. In all the experiments, the buffer size is set to be 20×16384 bits. The original Test Model 5 rate control procedure that is included in the encoder is also used for comparison purposes.

The average PSNR and computation complexity for the Football and Miss America sequences is presented in Table 3.2. Figure 3.5 (a), (b) shows the PSNR and buffer occupancy for the Football sequence at 1.152 Mbps. The PSNR curve is presented in display order, while the buffer occupancy is in coding order. From the figure, we observe higher PSNR for the steepest descent method. It is still better even when we compare the steepest descent method at GOP size 6 with the Test Model 5 at GOP size 12. Also, when the squared difference of MSE, $E(\mathbf{q})$, is introduced in the cost function, the PSNR curve becomes smoother, thus resulting in a more stable video quality. Figure 3.5 (c), (d) shows a similar result for the Miss America sequence at 1.152 Mbps. The effect of the weighting coefficient w is presented in Figure 3.6 (a), (b), and finally, a test case for steepest descent method with GOP size 12 is shown in Figure 3.6 (c), (d).

Football at 1.152 Mbps					
	Test Model 5		Steepest Descent		
GOP Size	12	6	6	6	6
Weighting Factor (w)	$\times$	$\times$	0	10^3	10^6
PSNR (dB)	30.6895	30.5728	31.1776	30.8281	30.7809
Complexity	1	1	28.8	56.8	51.1
Complexity for Last GOP	1	1	22	26	22

Miss America at 1.152 Mbps					
	Test Model 5		Steepest Descent		
GOP Size	12	6	6	6	6
Weighting Factor (w)	$\times$	$\times$	0	10^3	10^6
PSNR (dB)	41.4331	41.3376	42.1133	42.0891	42.1121
Complexity	1	1	18.1	15.9	15.0
Complexity for Last GOP	1	1	20	14	14

Miss America at 0.8 Mbps					
	Test Model 5		Steepest Descent		
GOP Size	12	6	6	6	6
Weighting Factor (w)	$\times$	$\times$	0	10^3	10^6
PSNR (dB)	40.8956	40.7740	41.4301	41.3819	41.3819
Complexity	1	1	25.5	31.7	31.7
Complexity for Last GOP	1	1	24	21	21

Table 3.2: MPEG encoding results.

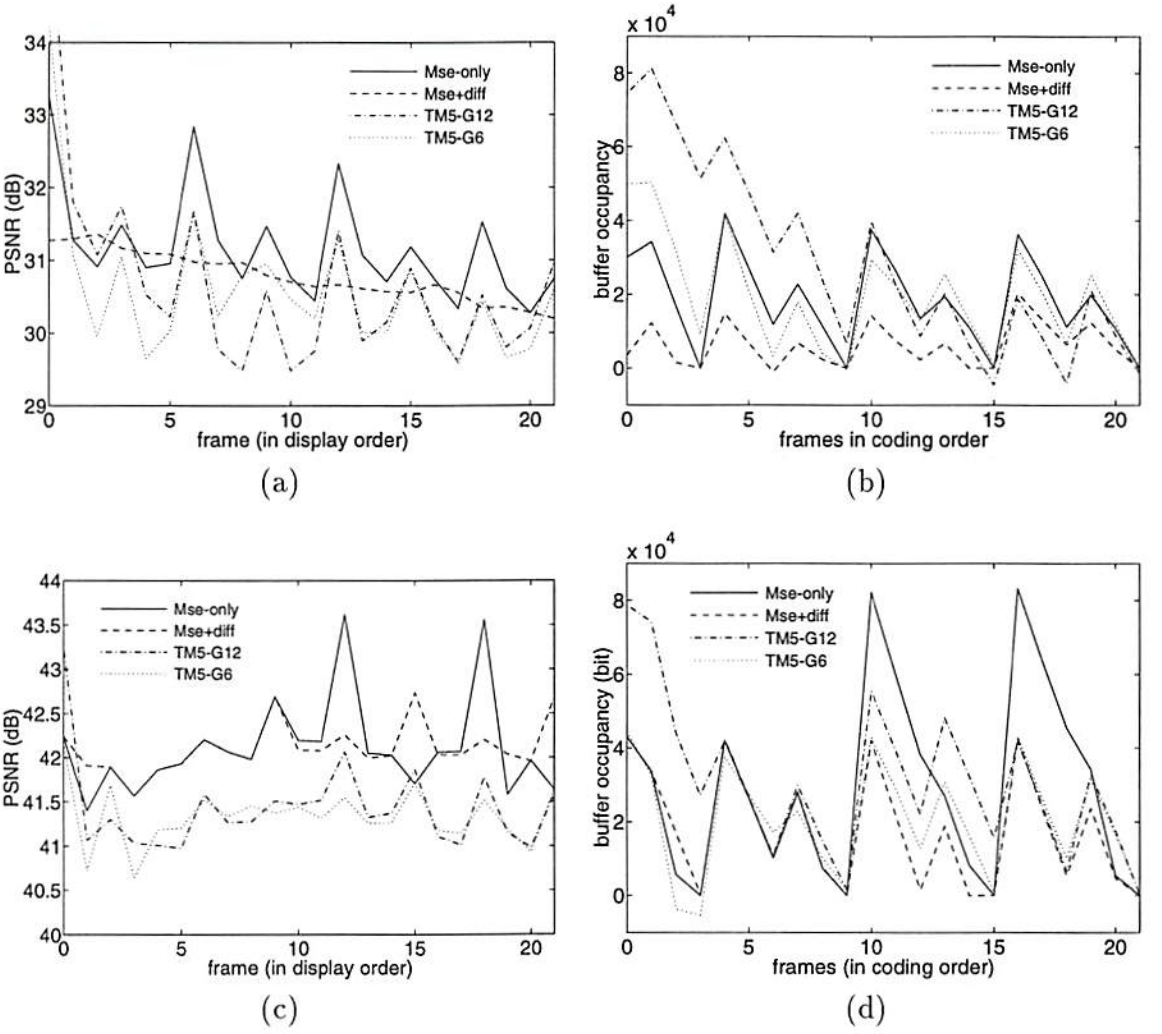


Figure 3.5: (a) PSNR, (b) buffer occupancy of Football sequence; (c) PSNR, (d) buffer occupancy of the Miss America sequence. The bit-rate is set to 1.152 Mbps for both sequence. In each figure, *MSE-only*: Steepest descent method with $w = 0$ and GOP size = 6; *MSE+diff*: Steepest descent method with $w = 10^6$ and GOP size = 6; *TM5-G12*: Test model 5 with GOP size = 12; *TM5-G6*: Test model 5 with GOP size = 6.

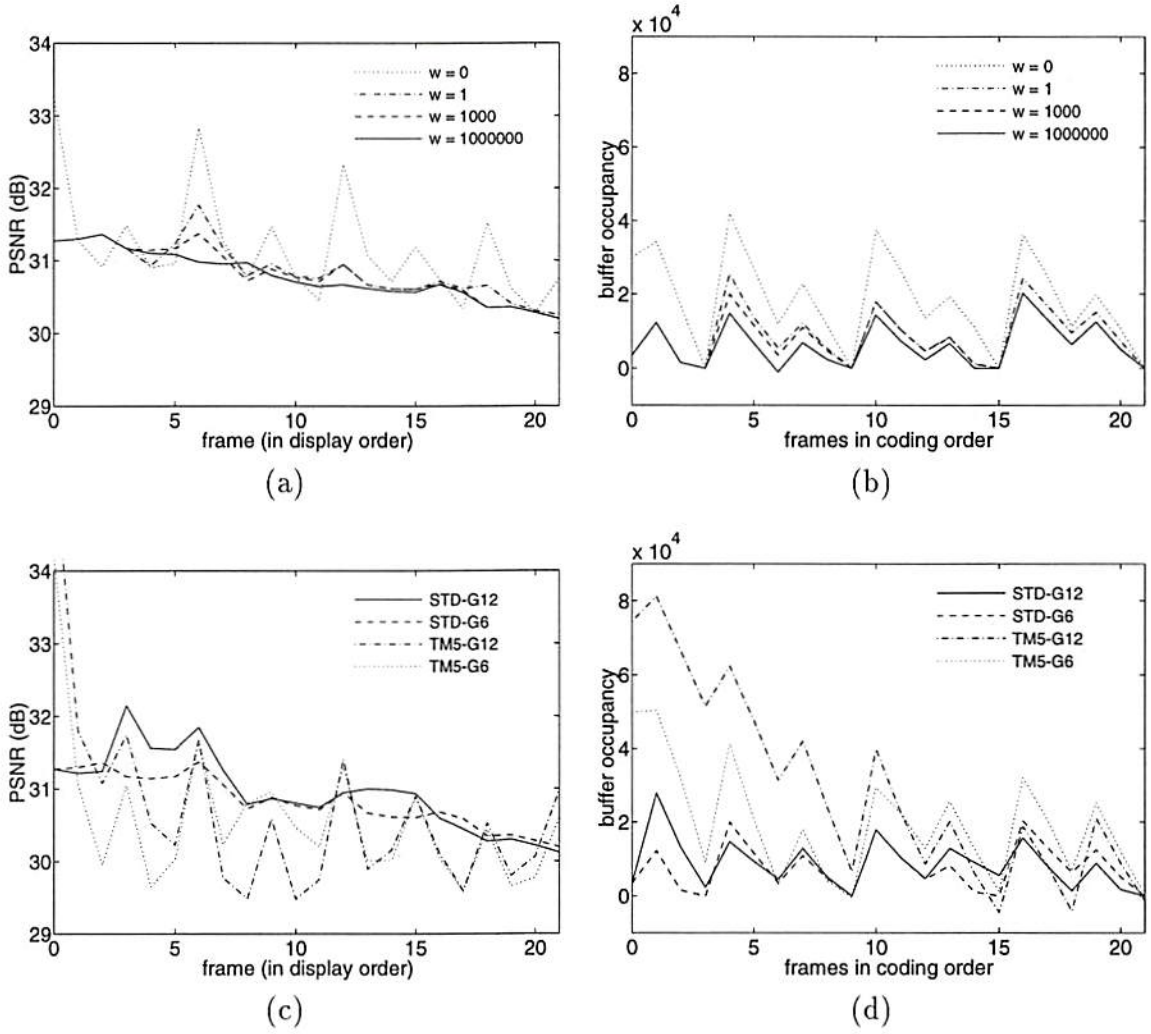


Figure 3.6: (a) PSNR, (b) buffer occupancy for Football sequence, over several different settings of w. These two figures shows the effect of the weighting coefficient, w. (c) PSNR, (d) buffer occupancy for Football sequence, *STD-G12*: Steepest descent with GOP size = 12; *STD-G6*: Steepest descent with GOP size = 6. *TM5-G12*; Test model 5 with GOP size = 12; *TM5-G6*: Test model 5 with GOP size = 6.

3.5 Conclusions

We have implemented and demonstrated the feasibility of using a gradient-based optimization algorithm for buffer control. By using this technique, we are able to achieve strictly constant bit rate per GOP, increase the overall quality (in MSE sense), while decreasing the variation of qualities between consecutive frames. From the results that we have reached, we would like to emphasize the following.

- Compared to TM5, we can observe the complexity is 30 - 50 times higher for the first few GOPs, where the initial guess is far away from the optimum solution. In the follow-up GOPs, because the solution in the previous GOP is used for the initial guess, the complexity is reduced to 15 to 25. The actual complexity can be made smaller by taking the costly motion estimation out of the iteration loop.

- We have observed that our technique using GOP of size 6 already has a better performance than Test Model 5 using GOP of size 12. When the GOP size is increased to 12 in our technique, the computation complexity becomes higher. This is due to the slow convergence in higher dimensional vector space. However, because only an I frame becomes P, the increase in PSNR not significant. A revised algorithm applicable to a higher GOP size will be presented in Chapter 5.

Even though a better quality (in the MSE sense) can be achieved by the gradient technique, it requires a higher computational complexity. This should not be a problem for benchmarking purposes. Further approximation can be done to make it suitable for off-line encoding and real-time encoding as described in the next chapter.

Chapter 4

Approximation of Rate-Distortion Functions

4.1 Introduction

Most of the computation cost for the gradient based rate control algorithm presented in Chapter 3 comes from the evaluation of rate-quantization and distortion-quantization functions, which involves repeatedly encoding the video data on all required quantization settings. In order to reduce the computations, it is necessary to introduce an approximation model so that we can estimate rate and distortion without having to encode with all settings.

As shown in Chapter 2, several rate and distortion modeling methods [53, 18, 5, 13, 25] have been proposed or used in their specific bit-rate control schemes, and most of them are based on the exponential statistics model. The difficulty in determining model parameters and high model errors make them unsuitable for

our optimization control problem. In this chapter, we present a novel approximation model which makes relatively few assumptions on the shape of the R-D characteristics and takes into account the typical dependency of video coding. These models are based on computing a few R-D points and interpolating the remaining data using spline functions [33, 34, 35].

This chapter is organized as follows. In Section 4.2 we describe the formulation of spline interpolation functions and apply this function to an optimal adaptive quantization algorithm for image compression. In Section 4.3 we present a scheme to model the coding dependencies for P and B frames. In Section 4.4 we apply the approximation model to the gradient-based rate control algorithm described in the previous chapter, and present several experimental results.

4.2 Spline Approximation Method

In a typical DCT-based encoder the rate-distortion trade-off is controlled by a quantization scale. This parameter is used to compute the step size of a set of uniform quantizers used for the DCT coefficients (see [44, 19, 38] for details). When an image block (or an entire image frame if constant quantization is used) is quantized and encoded with a specific quantization scale, q, the rate (the number of bits generated by the coder), $r(q)$, and the distortion (here the MSE is used), $d(q)$, can be calculated. The computational cost can be reduced significantly if these two function values can be correctly estimated without actually quantizing and encoding the source data. However, due to the complex nonlinear properties of the quantization and entropy coding processes, it is difficult to predict the function value accurately enough by using simple mathematical expressions. In

this dissertation, we propose an approach which calls for encoding the data and measuring the R-D functions, but *only on a small set of quantization scales* which we call "control points". Piece-wise polynomials, or splines, are then used to interpolate the function for other q's where the actual data has not been measured.

4.2.1 Formulation of Spline Interpolation Function

Two different interpolation functions are tested in this dissertation, linear interpolation and cubic interpolation. Linear interpolation simply uses a straight line to connect two consecutive control points. Its approximation error is expected to be higher when applied to the rate-quantization curves, because the shape of these functions cannot be approximated by line segments.

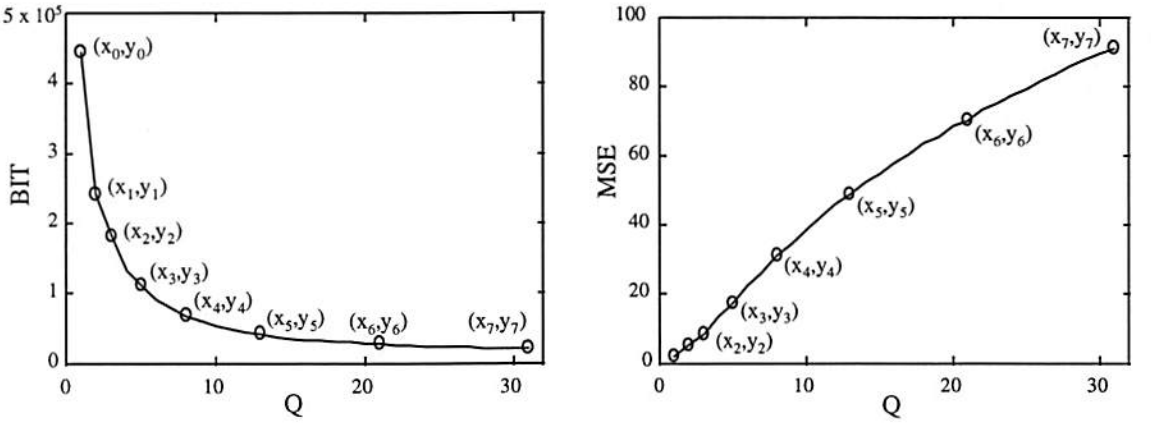
A second choice is to use cubic interpolation. Because the rate and distortion functions are to be used in an optimization algorithm (gradient search, Lagrange method, etc.), it may be useful to operate with interpolation functions having well-defined first order derivatives. A good candidate would be the "interpolating cubic-spline", which possesses the second-order continuity property (see Chapter 4 of [12]). The disadvantage of this method is that the interpolation polynomial for any given segment (a segment is defined as a set of points between the two consecutive control points) depends on all the control points, i.e., it will require the coder to encode the source on all the control points even though only a small portion of the function data is required in the rate control algorithm. In this dissertation, we use another type of spline, which requires smaller computation cost, still possesses first-order continuity, and for which each segment depends only on four nearest control points.

We assume the control points are defined as (x_i, y_i), $i = 1, 2, \ldots, M$, where M

is total number of control points. Fig. 4.1 shows an example set of control points, where x_i represents the quantization scale (for MPEG, the applicable values are $\{1, 2, \ldots, 31\}$), and y_i represents the actual measured rate or distortion. The function between two consecutive control points, x_i and x_{i+1}, is defined as

$$f_i(x) = a_i \cdot x^3 + b_i \cdot x^2 + c_i \cdot x + d_i \tag{4.1}$$

where $i = 1, 2, \ldots, M-1$. There are $M-1$ polynomials, each corresponding to one segment. For each polynomial, the four parameters, a_i, b_i, c_i, d_i, can be derived from four control points, (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}), by imposing the following two constrains:



(a) Bits (b) MSE

Figure 4.1: Control points for a typical (a) rate and (b) distortion curve. In the figure, a control point (x_i, y_i) represents that if the quantization scale is set to x_i, the measured rate or distortion value is y_i.

1. The interpolated data should take the same values as the original one at the control points, hence

$$x_i^3 \cdot a_i + x_i^2 \cdot b_i + x_i \cdot c_i + d_i \;=\; y_i, \tag{4.2}$$

$$x_{i+1}^3 \cdot a_i + x_{i+1}^2 \cdot b_i + x_{i+1} \cdot c_i + d_i = y_{i+1}. \quad (4.3)$$

2. The first-order derivative should be continuous at the control points. This condition can be achieved by defining the slope at control point x_i as (the first derivative of $f(x)$ is denoted as $f'(x)$)

$$f_i'(x_i) = f_{i-1}'(x_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}. \quad (4.4)$$

By taking the derivative of (4.1) and substituting into (4.4) on the two end points of $f_i(x)$, we get

$$3x_i^2 \cdot a_i + 2x_i \cdot b_i + c_i = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}, \quad (4.5)$$

$$3x_{i+1}^2 \cdot a_i + 2x_{i+1} \cdot b_i + c_i = \frac{y_{i+2} - y_i}{x_{i+2} - x_i}. \quad (4.6)$$

The four unknowns of $f_i(x)$, a_i, b_i, c_i, d_i, can be readily found by solving the set of equations (4.2), (4.3), (4.5), and (4.6). The computation can be simplified further if we translate x_i to 0, and x_{i+1} to 1, by using

$$z(x) = \frac{x - x_i}{x_{i+1} - x_i}, \quad (4.7)$$

and re-define the polynomial defined as $f_i(z) = a_i \cdot z^3 + b_i \cdot z^2 + c_i \cdot z + d_i$. Now, the four parameters, a_i, b_i, c_i, d_i can be calculated directly using

$$d_1 = (x_{i+1} - x_i) \cdot \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}, \quad (4.8)$$

$$d_2 = (x_{i+1} - x_i) \cdot \frac{y_{i+2} - y_i}{x_{i+2} - x_i}, \quad (4.9)$$

$$e = y_{i+1} - y_i - d_1, \tag{4.10}$$

$$g = d_2 - d_1, \tag{4.11}$$

$$a_i = g - 2e, \tag{4.12}$$

$$b_i = 3e - g, \tag{4.13}$$

$$c_i = d_i, \tag{4.14}$$

$$d_i = y_i. \tag{4.15}$$

4.2.2 Compliance Test for Intra-Frame Approximation

In order to capture the exponential-decay property, which is typically observed in rate-quantization functions, we choose the control points to be with the relation as $x_i = x_{i-1} + x_{i-2}$, so that, in the MPEG case, the set of eight control points becomes

$$\{1, 2, 3, 5, 8, 13, 21, 31\}.$$

However, on a typical video sequence at standard rate (e.g. CIF at 1.152 Mbps), the settings for $q = 1, 2$, or even $q = 3, 4$, are rarely used, hence, only 5 to 6 control points are required in most cases. Note that while approximately exponential characteristics are typical, the error incurred with our approach will normally be smaller than using a exponential model, because we have more degrees of freedom, and the characteristics are not exactly exponential.

To test the accuracy of approximation at frame level, we use an MPEG-2 encoder [41] to encode the image frames from several different sequences and different frame type, by encoding and measuring the MSE and code length for all possible quantization settings (from 1 to 31). Note that when encoding P and B

frames, the quantization scales of their reference frames are set to 10. Also note that the computational overhead is small because the DCT operation only has to be done once, and only the quantizations and the zero-run-length variable-length-codings are involved in each iteration of the coding process. Then, based on the value at pre-defined control points, we derive the approximated data. Fig. 4.2 shows the rate of an I frame in the football sequence, as a function of q. The relative errors are then derived in the same way as in Section 2.5, using (2.27). The results are shown in Table 4.1. For convenience, the data from the optimum exponential model in Table 2.1 are also included for comparison. The data shows that the interpolation approximation offers smaller approximation errors than the optimum exponential model, and cubic interpolation is better than linear interpolation, especially for the rate-quantization functions.

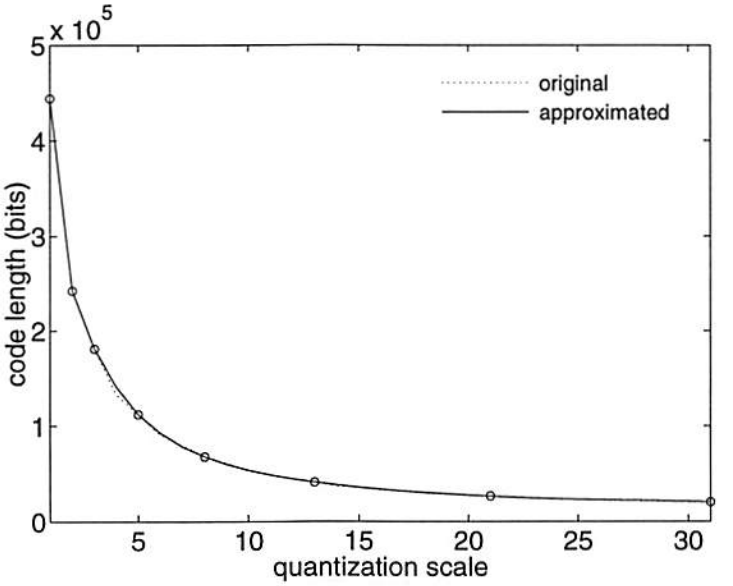


Figure 4.2: Rate function of an I frame in the football sequence. The circles indicate the control points, which are chosen to capture the exponential-decay property of the rate function.

Intra-frames Average Relative Errors for BITS(q)			
	opt.expon	pw.linear	pw.cubic
Claire	5.77%	2.27%	0.65%
Football	14.95%	4.30%	0.90%
Miss America	26.75%	3.07%	1.16%
Susie	21.15%	3.35%	1.24%

Intra-frames Maximum Relative Errors for BITS(q)			
	opt.expon	pw.linear	pw.cubic
Claire	28 46%	28.89%	7.57%
Football	77.09%	15.43%	6.32%
Miss America	100.03%	26.05%	9.86%
Susie	68.00%	23.65%	7.13%

Intra-frames Relative Errors for MSE(q)				
	average error		maximum error	
	pw.linear	pw.cubic	pw.linear	pw.cubic
Claire	0.72%	0.55%	2.95%	4.37%
Football	0.95%	0.55%	6.13%	7.01%
Miss America	0.65%	0.43%	4.63%	3.84%
Susie	0.84%	0.51%	5.95%	6.10%

Table 4.1: Relative errors for intra-frame approximation functions. The statistics are over the entire quantization scale range, and over three type of frames (I, P, B). opt.expon: optimum exponential. pw.linear: piecewise linear. pw.cubic: piecewise cubic.

4.2.3 Application to Local Adaptive Quantization

By applying the spline approximation model at DCT block level, we are able to reduce the computational load in searching for optimal adaptive quantization in DCT-based image compression. Suppose there are M_b blocks in an image, and for each block the rate and distortion for a given quantization setting are denoted as $r_i(q_i)$ and $d_i(q_i)$, where i is the index for block and q_i is the quantization scale (an integer between 1 and 25 in our image-coding experiment) assigned to block i. The optimum adaptive quantization problem is to determine quantization scales

for all blocks $(q_1, q_2, \ldots, q_{M_b})$, such that the overall distortion

$$\sum_{i=1}^{M_b} d_i(q_i) \tag{4.16}$$

is minimized for variables $(q_1, q_2, \ldots, q_{M_b})$, subject to the rate constrains (total bit-budget is B)

$$\sum_{i=1}^{M_b} r_i(q_i) \leq B, \tag{4.17}$$

which is the same as in Formulation 2.1. Because there is no inter-block dependency, each variable can be optimized independently and thus the problem can be solved by the Lagrange multiplier method of Algorithm 2.1. Note that the overhead for coding the quantization scale for each block is ignored in the above formulation (but is included in counting the total bits). That overhead can be taken into account by using the techniques proposed in [42].

In the experiment, we encode the 512 × 512 grayscale Lena image using a modified JPEG encoder. The modification was made such that a quantization scale can be assigned for each DCT block as is done in MPEG. To test the effectiveness of the spline approximation model, we replace $r_i(q_i)$ and $d_i(q_i)$ by the approximated data, and run the adaptive quantization procedure again. The results are shown in Table 4.2 and Fig. 4.3. We conclude from the results that, (i) the spline approximated model produces a much smoother R-D curve, which may have potential to be used to reduce the complexity of the search procedure in optimization algorithm, (ii) the result with approximated R-D is close to the one using actual R-D, and (iii) the constraint (4.17) may not be strictly satisfied due to the error in the approximated rate. In practice, this will not be a problem because the errors are typically small and the variations can be absorbed through

buffering. By using the spline model, the computation complexity for the evaluating R-Ds is reduced to about 20% (6 control points instead of 31 settings). Fig. 4.4 shows the original and spline interpolated rate-distortion function of a DCT block. Note also that many blocks have R-D characteristics similar to that of Fig. 4.4 and the lack of smoothness in the shape makes our approach based on several control points more effective than exponential based models.

constant q.			a.q. with original R-D		a.q. with spline R-D	
q	bpp	PSNR	bits overflow	PSNR	bits overflow	PSNR
3	1.554	37.96	-34	38.66	1223	38.59
6	0.993	35.94	-17	36.58	2290	36.46
9	0.755	34.84	-2	35.32	2358	35.16
12	0.621	34.08	0	34.37	2095	34.26

Table 4.2: Adaptive quantization encoding of Lena image. *constant q.*: constant quantization; *a.q. with original R-D*: adaptive quantization using the original R-D; *a.q. with spline R-D*: adaptive quantization using spline approximated R-D. The *bits overflow* is the difference in bits between the actual number of bits generated and the bit-budget.

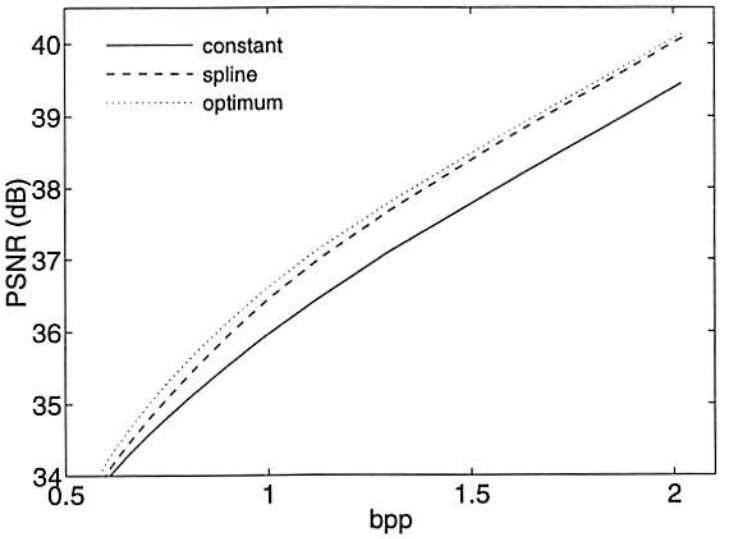


Figure 4.3: PSNR curves of constant quantization, adaptive quantization using original data, and adaptive quantization using spline approximated data.

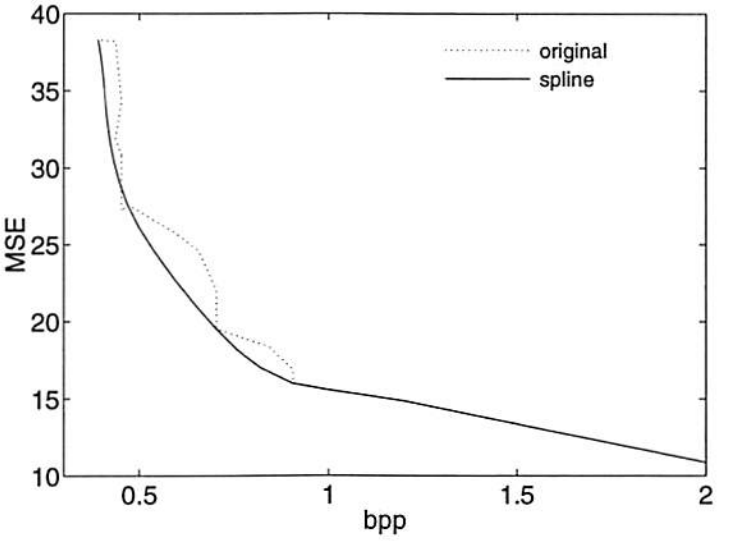


Figure 4.4: Comparison between the original and approximated rate-distortion function of a DCT block in the Lena image, to illustrate the lack of smoothness in the original R-D characteristics.

4.3 Inter-Frame Dependency Model

We now consider the P and B frames in MPEG video. The intra-frame approximation functions defined in the previous section can be applied when their reference frames are fixed. However, because the R-D characteristics also depend on the quality of their reference frames, we have to deal with multi-dimensional functions. In this section, we introduce methods that are less complex than full-blown multi-dimensional models while still capturing the inter-frame dependencies. Note that the ideas presented in this section are introduced in an MPEG framework, but are applicable to more general video coding environments. To simplify the computation, we make the motion estimation refer to the original image, so that it only has to be computed once for each of the P and B frames, and does not have to be re-computed when the P and B frames are encoded repeatedly to sample their respective R-D functions at the control points.

4.3.1 R-D Model of Predictive Frames

We consider the first P frame in a GOP, and its reference I frame (note that the model still can be applied when the reference is another P frame). Because of the dependency, the rate and distortion functions become two-dimensional, i.e., they have the form of $d(q_I, q_P)$ and $r(q_I, q_P)$, where q_I and q_P are the quantization scales for the I and P frame respectively. Fig. 4.5 shows the function $d(q_I, q_P)$ of a P frame from the Football sequence as a 2-D surface plot. Note the dependency, so that the data has to be sampled in the two-dimensional space. One straightforward extension is to sample the data at the same 6 control points as in Section 4.2 for each dimension (total 36 control points), but this requires many more computations. This is because, in order to compute the data for each additional control point along q_I axis, the I frame has to be re-compressed and reconstructed again (involving DCT, quantization, de-quantization, and IDCT), and the P frame has to be re-encoded (involving prediction, DCT, quantization, and encoding). This complexity is much higher than the one for computing the data along q_P axis (only involving quantization and encoding for the P frame). To cope with this problem, we introduce a model for inter-frame dependency which only requires two control points along q_I axis.

To consider the effect of prediction, we denote the data in an I frame as X_1, and its quantized version as $\hat{X}_1 = Q_I(X_1)$, as shown in Fig. 4.6. We also denote the data in a P frame as X_2, so that the prediction residue becomes $D = X_2 - Q_I(X_1)$. Note that the motion compensation block in Fig. 4.6 is ignored in the expressions because it only affects the coordinate index but not the signal itself. Hence, the

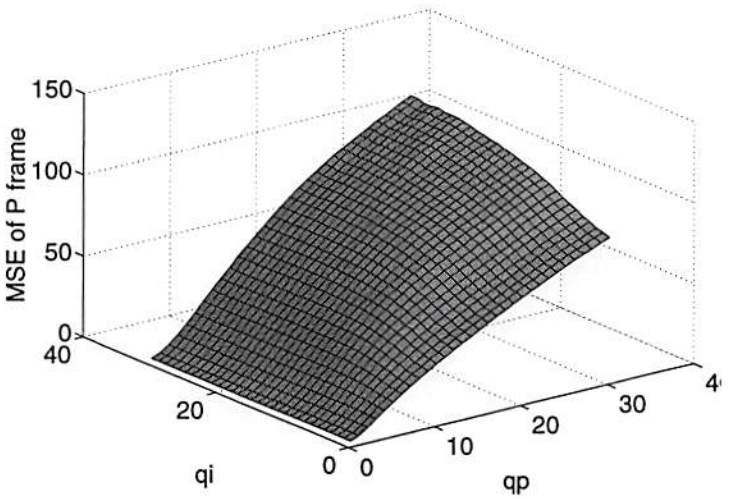


Figure 4.5: MSE of a P frame in the Football sequence, plotted as a 2-D function of q_I and q_P.

reconstructed P frame becomes

$$\hat{X}_2 = Q_I(X_1) + Q_P\left(X_2 - Q_I(X_1)\right). \tag{4.18}$$

Now we denote the error for the I frame as $E_I = X_1 - Q_I(X_1)$, and the error for the P frame as $E_P = X_2 - \hat{X}_2$. Substituting $Q_I(X_1)$ in (4.18) by $X_1 - E_I$, we get

$$\hat{X}_2 = X_1 - E_I + Q_P(X_2 - X_1 + E_I), \tag{4.19}$$

and the error for P frame becomes

$$E_P = X_2 - X_1 + E_I - Q_P(X_2 - X_1 + E_I). \tag{4.20}$$

Now consider the case where q_I *is smaller than* q_P, which means $u_I < u_P$, where u_P and u_I are, respectively, the quantization step-size of the uniform quantizer for P-frame and I-frame. If the motion prediction is correct and the initial

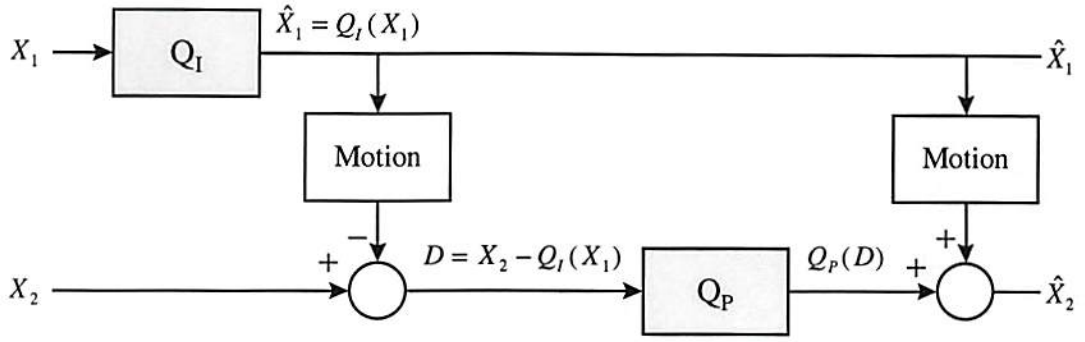


Figure 4.6: Prediction Model

error was small we have $Q_p(X_2 - X_1 + E_I) \approx 0$, due to the coarse quantization used for the P frame. Thus we have

$$E_P \approx X_2 - X_1 + E_I, \tag{4.21}$$

and, in this case, for a fixed q_P, when q_I varies, we expect a linear relationship between the error for I and P frames. For the case where blocks are perfectly matched to their predicted blocks from the reference frame, or, $X_2 \approx X_1$, the quantization can be illustrated as a two-step quantizer shown in Fig. 4.7. So, it is reasonable to assume that the quantization error in the predictive frame linearly depends on the error in its reference frame if $q_I < q_P$.

In the second case where q_I *is larger than* q_P, i.e., $u_I > u_P$. we have a finer quantizer u_P to quantize the $X_2 - X_1 + E_I$ term in (4.20). Hence the quantization error is dominated by u_P, i.e., the maximum absolute value of E_P is not larger than $u_P/2$, and independent of u_I. So, in this case, for a fixed q_P, when q_I varies, we expect E_P to be roughly constant. Again for the case where blocks are perfectly matched to their predicted blocks from the reference frame, or, $X_2 \approx X_1$, the

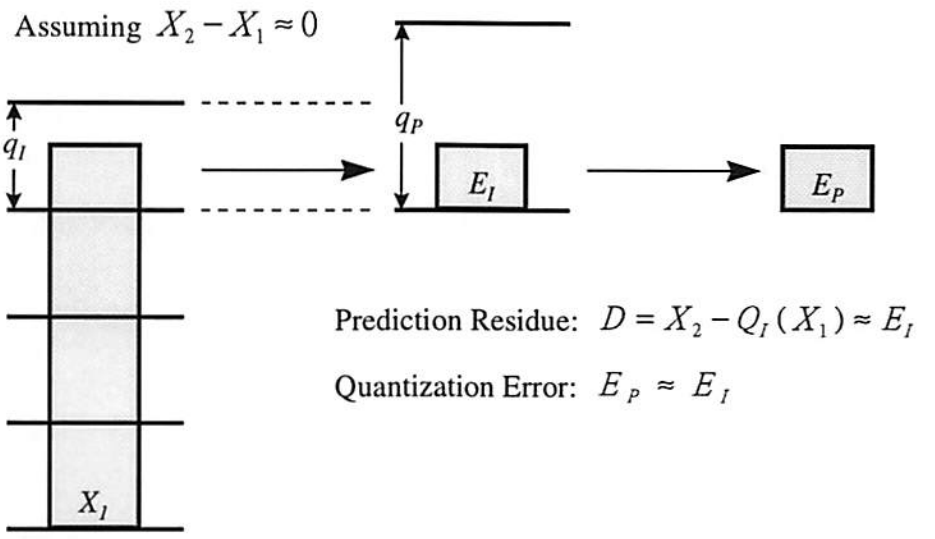


Figure 4.7: Quantization of predictive signal for $q_I < q_P$

quantization can be illustrated as a two-step quantizer shown in Fig. 4.8, which shows that the error in the P frame is independent of q_I. On the other hand, in the case where good inter-frame prediction can not be made, e.g., for high activity, fast moving video sequence, another factor which affects the relationship between E_P and E_I comes from the selection of intra and inter block coding in MPEG. If the MSE in the reference is larger, not only the rate and distortion of the predictive frame will become larger, but also more macroblocks will be coded in "intra" mode (given the typical decision rules used in general MPEG encoders, e.g. in [41]), which will also decrease the dependency on the reference frame. Thus it is reasonable to assume that, for any video sequence, the quantization error in the predictive frame is independent of its reference frame, if $q_I > q_P$.

The two cases shown in Fig. 4.7 and Fig. 4.8 can be summarized by a single

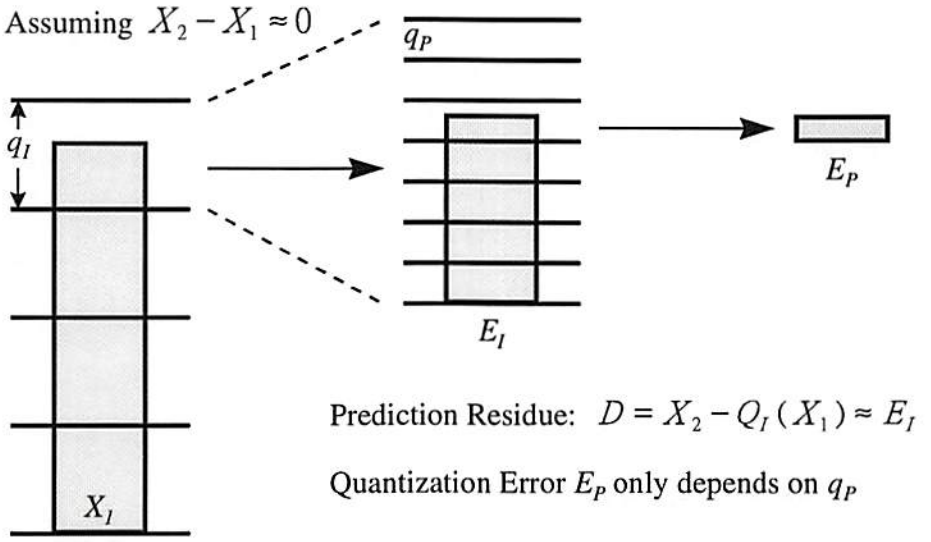


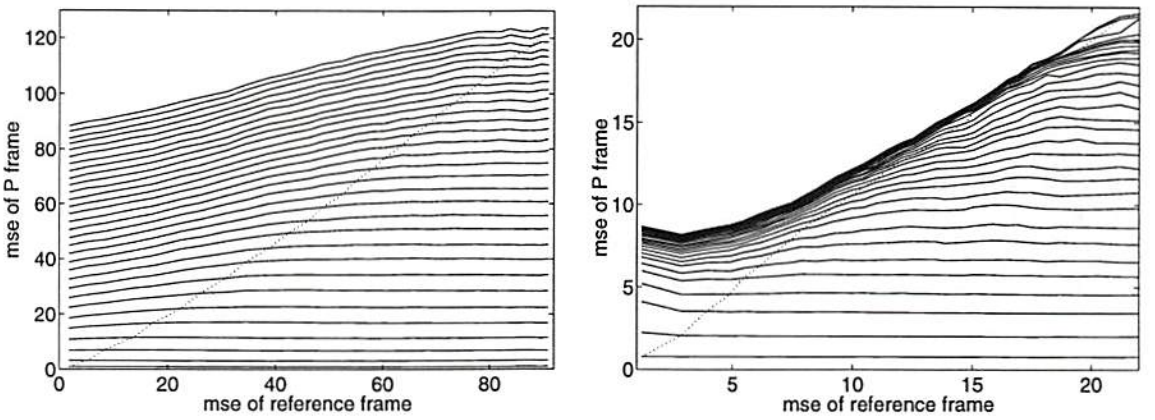
Figure 4.8: Quantization of predictive signal for $q_I > q_P$

statement, i.e., the R-D characteristics only depend on

$$\min(q_I, q_P). \tag{4.22}$$

For the intermediate cases, where q_I and q_P have similar values, we expect the function of E_P vs. E_I will be gradually changing from linear-increasing to constant.

Based on the above analysis, we conclude that, for a fixed q_P, if q_I is increasing from a small value, we should expect the MSE of P frame to be a linear increasing function of the MSE of I frame. The dependency will decrease when q_I is nearly equal to q_P, and after that, it will be completely independent of the I frame and become a constant value. Fig. 4.9 shows the results derived by repeatedly encoding the I and P frames with all the quantization settings. The experimental results confirm the analysis.



(a) Football sequence (b) Miss America sequence

Figure 4.9: MSE for the P frames from two video sequences, plotted as a function of MSE for their reference frames. Each solid line is a MSE curve for a given q in the predictive frame. The dotted line indicates the boundary where q for the predictive and reference frames are equal.

4.3.2 Approximation Models

Distortion Functions for P Frames

The I-P dependency model is proposed as below. Suppose q_P fixed at a constant value C, so that $d(q_I, q_P = C)$ becomes a one-dimensional function with variable q_I. The MSE of the reference frame (I-frame) is denoted as $d_I(q_I)$. The frame dependency for the distortion of a P-frame is modeled as a linear increasing function with respect to $d_I(q_I)$ for $q_I \leq C$, and as a constant function for $q_I > C$, as shown in the following expression:

$$d(q_I, C) = \begin{cases} \alpha - \beta \cdot [d_I(C) - d_I(q_I)], & q_I \leq C, \\ \alpha, & q_I > C. \end{cases} \tag{4.23}$$

The two model parameters, α and β, can be determined by encoding and measuring the distortion at two values of q_I, as shown in Fig. 4.10, which is done by

the following.

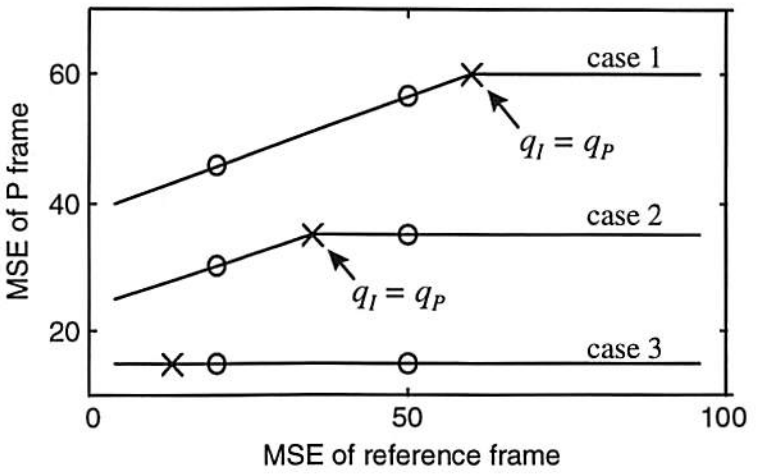


Figure 4.10: Constant-linear function reconstructed by two control points.

We denote the position of the two control points as x_1 and x_2, and the measured function values at control points as y_1 and y_2, respectively. For case 1 in Fig. 4.10 where $x_2 < C$, both control points fall on the linear-increasing part. In this case, we substitute the values of the two control points into (4.23) and solve for α and β, and get

$$\alpha = y_1 + (y_2 - y_1) \cdot \frac{d_I(C) - d_I(x_1)}{d_I(x_2) - d_I(x_1)}, \tag{4.24}$$

$$\beta = \frac{y_2 - y_1}{d_I(x_2) - d_I(x_1)}. \tag{4.25}$$

For case 2 in Fig. 4.10 where $x_1 < C < x_2$, the second control point x_2 falls on the constant part, hence we have

$$\alpha = y_2, \tag{4.26}$$

$$\beta = \frac{y_2 - y_1}{d_I(C) - d_I(x_1)}. \tag{4.27}$$

For case 3 in Fig. 4.10 where $C < x_1$, both control points fall on the constant part. It is not possible to solve for α and β in this case. Because C (value of q_P) is small, it is reasonable to assume that the entire function is constant. However, because the value of y_1 and y_2 may be different, we take this difference into account and use the following piece-wise linear expression to approximate this case:

$$d(q_I, C) = \begin{cases} d(x_1, C), & q_I \leq x_1, \\ \frac{d(x_1,C)(d_I(x_2)-d_I(q_I))+d(x_2,C)(d_I(q_I)-d_I(x_1))}{d_I(x_1)-d_I(x_2)}, & x_1 < q_I < x_2, \\ d(x_2, C), & q_I \geq x_2. \end{cases} \tag{4.28}$$

The complete reconstruction of the two-dimensional function $d(q_I, d_P)$ is illustrated in Fig. 4.11. The values of the two inter-frame control points are chosen to be 5 and 13, and the same spline model with 6 control points as in Section 4.2 is used along q_P axis. Thus the set of 12 control points becomes

$$\left\{ \begin{array}{cccccc} (5,3) & (5,5) & (5,8) & (5,13) & (5,21) & (5,31) \\ (13,3) & (13,5) & (13,8) & (13,13) & (13,21) & (13,31) \end{array} \right\}. \tag{4.29}$$

To interpolate the function value for any given settings, say $(10, 10)$, the above inter-frame model is applied 4 times with C set to $\{5, 8, 13, 21\}$, so that the function values are derived at $(10, 5)$, $(10, 8)$, $(10, 13)$ $(10, 21)$. Then spline interpolation is used to derive the value at $(10, 10)$ using the 4 derived data.

Rate Functions for P Frames

Due to the difference in properties, a similar model for distortion does not work as well for the rate. Fig. 4.12 shows the inter-frame dependency of rates for P

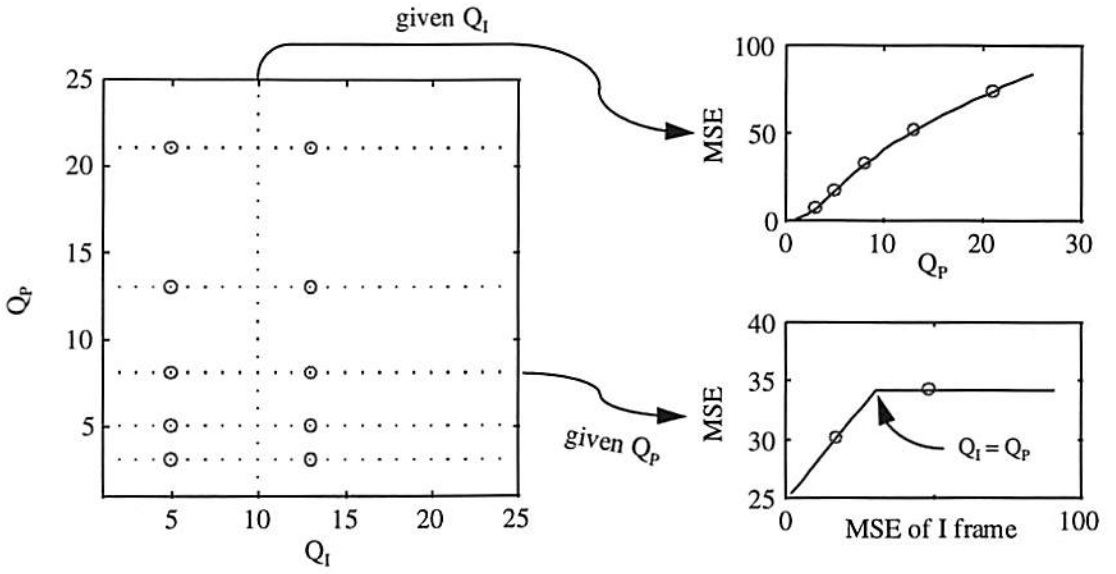
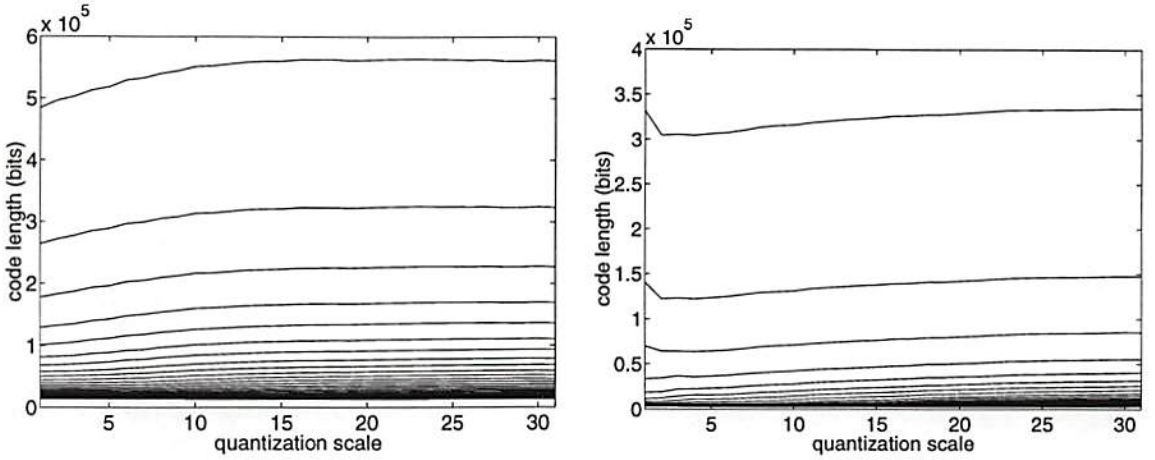


Figure 4.11: Reconstruction of approximated distortions of P frames. The left diagram shows the 2-D domain of a P frame distortion function, $d(Q_I, Q_P)$. The circles in the diagram indicate the control points, at which actual function values are sampled. To reconstruct the function values from these control points, we first approximate the function values along the horizontal (Q_I) direction where control points are available (indicated by horizontal dashed lines) using the inter-frame model, and then approximate the values along the vertical (Q_P) direction with the intra-frame interpolation functions.

frames in the Football and Miss America sequence. From the figure and based on the observation over several other video sequences, we conclude that, for the quantization scales between 3 and 24, the inter-frame dependency for rate is reasonably low. Because we already have two known values from the two control points, it is reasonable to simply use the same linear interpolation model as in (4.28) to approximate the rate:

$$r(q_I, C) = \begin{cases} r(x_1, C), & q_I \leq x_1, \\ \frac{r(x_1,C)(d_I(x_2)-d_I(q_I))+r(x_2,C)(d_I(q_I)-d_I(x_1))}{d_I(x_1)-d_I(x_2)}, & x_1 < q_I < x_2, \\ r(x_2, C), & q_I \geq x_2. \end{cases} \tag{4.30}$$



(a) Football sequence (b) Miss America sequence

Figure 4.12: Code length for the P frames from two video sequences, plotted as a function of quantization scales for their reference frames. Each solid line is a code length curve for a given q in the predictive frame.

Rate and Distortion Functions for B Frames

For B frames, the distortion function becomes $d(q_I, q_P, q_B)$, where q_B is the quantization scale for the B frame itself, and q_I and q_P are the quantization scales for the two reference frames. With a straightforward extension of (4.22), it is reasonable to assume that the R-D characteristics only depend on

$$\min(q_I, q_P, q_B). \tag{4.31}$$

Based on this assumption and also to keep the computation simple, as illustrated in Fig. 4.13, we first fix one reference frame by setting $q_I = c$ (where c is one of the inter-frame control points), and evaluate the dependency for the other reference frame by using the same model for P frames to get $d_1(c, q_P, q_B)$. We then fix the other reference frame and derive $d_2(q_I, c, q_B)$. Finally, $d(q_I, q_P, q_B)$ is defined as $\min(d_1(c, q_P, q_B), d_2(q_I, c, q_B))$. This procedure also simulates in part the strategy

for selecting "forward" or "backward" motion vectors in the MPEG encoder. The same approach is also used for the rate function. There are a total of 18 control points to be measured if the same set of control points as in (4.29) is used, as

$$\left\{ \begin{array}{cccccc} (5,5,3) & (5,5,5) & (5,5,8) & (5,5,13) & (5,5,21) & (5,5,31) \\ (5,13,3) & (5,13,5) & (5,13,8) & (5,13,13) & (5,13,21) & (5,13,31) \\ (13,5,3) & (13,5,5) & (13,5,8) & (13,5,13) & (13,5,21) & (13,5,31) \end{array} \right\}. \quad (4.32)$$

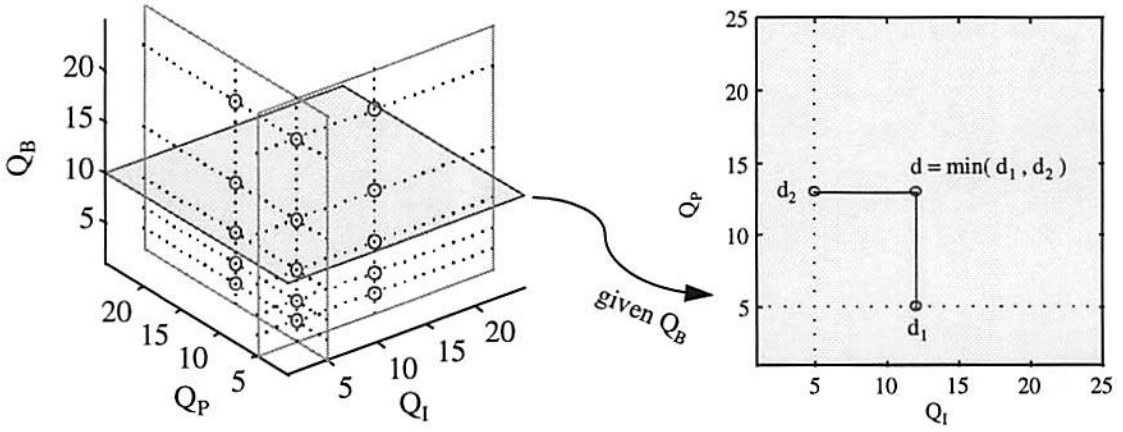


Figure 4.13: Reconstruction of approximated distortions of B frames. The left diagram shows the 3-D domain of a B frame distortion function. The circles in the diagram indicate the control points, at which actual function values are sampled. To reconstruct the function values from these control points, we first approximate the function values on the two vertical planes (shown in the left diagram) using the same procedures as for P frames. Then, for any given point in the space, the two values of its perpendicular projections on the two planes are picked and the smaller one is chosen to be the approximated function value, as shown in the right diagram.

4.3.3 Compliance Test for Inter-Frame Model

We use a MPEG-2 encoder [41] to test the accuracy of the approximation model, with the following steps. We first encode the frames, measure and record their

MSE and code length for every possible quantization settings. Based on the function values at the pre-defined control points (i.e., $\{1, 2, 3, 5, 8, 13, 21, 31\}$ for intra-coded frame approximation, $\{5, 13\}$ for inter-frame model), we build a model using the procedure described in Section 4.2 and Section 4.3, and calculate the estimated rate and distortion values. The relative errors are then calculated by (2.27).

The average and maximum of relative errors are calculated over a typical operating range of quantization scales, which is from 3 to 24. The results are shown in Table 4.3 and Table 4.4. The results show that the errors are reasonably small for P frames but somewhat larger for B frames. Several sample graphical comparisons are also shown in following figures. Fig. 4.14 shows the 3-D plot of the approximated distortion function of a P frame in the football sequence, for comparison with the original data shown in Fig. 4.5. Fig. 4.15 shows a case of the MSE of a P frame in the football sequence, as a function of MSE its reference frame. Fig. 4.16 shows MSE of a B frame in the football sequence, as a 2-D function of q's for the two reference frames. Although there are still errors introduced by the approximation models, the actual usefulness of the methods comes from using them in the optimization framework for rate control, which will be presented in the next section.

	P-Frame BITS Model Errors			
	average error		maximum error	
	linear	cubic	linear	cubic
Claire	9.34%	2.49%	40.90%	33.02%
Football	5.38%	0.66%	14.01%	8.41%
Miss America	12.39%	3.27%	43.54%	45.82%
Susie	11.04%	2.92%	39.33%	15.88%

	P-Frame MSE Model Errors			
	average error		maximum error	
	linear	cubic	linear	cubic
Claire	3.03%	0.88%	12.30%	12.30%
Football	1.81%	0.39%	5.59%	6.60%
Miss America	2.89%	0.89%	11.03%	11.03%
Susie	4.30%	1.24%	15.88%	15.88%

Table 4.3: Relative errors for predictive coding model. The statistics are calculated over the range from 3 to 24.

	B-Frame BITS Model Errors			
	average error		maximum error	
	linear	cubic	linear	cubic
Football B1	5.08%	3.74%	22.14%	22.72%
Football B2	5.73%	4.43%	23.43%	23.64%

	B-Frame MSE Model Errors			
	average error		maximum error	
	linear	cubic	linear	cubic
Football B1	3.08%	2.61%	17.56%	17.56%
Football B2	3.13%	2.58%	13.92%	14.73%

Table 4.4: Relative errors for bi-directional predictive coding model. The statistics are calculated over the range from 3 to 24.

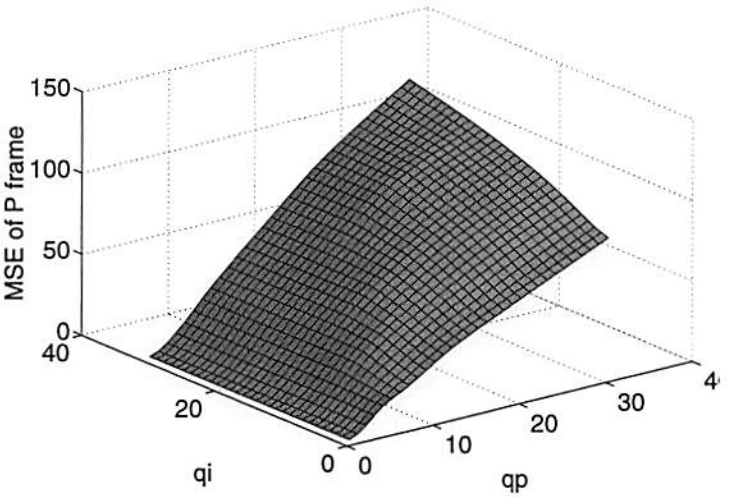


Figure 4.14: Reconstructed function of MSE of a P frame in the Football sequence using the P approximation model, plotted as a 2-D function of q_I and q_P.

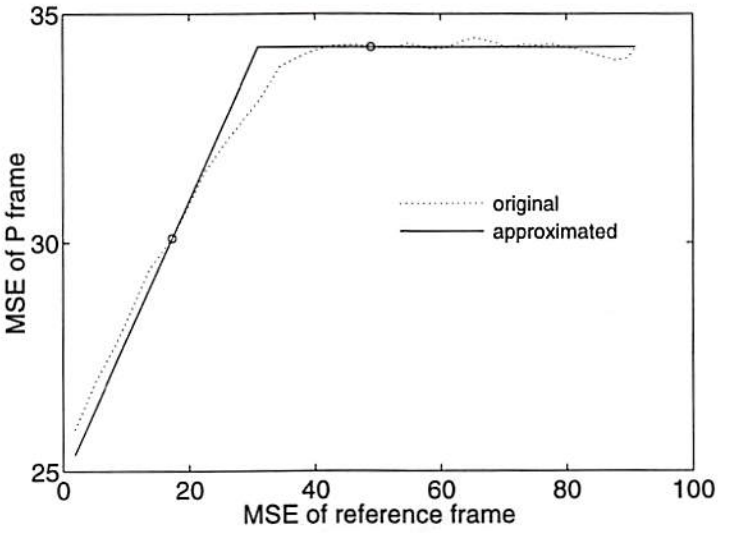
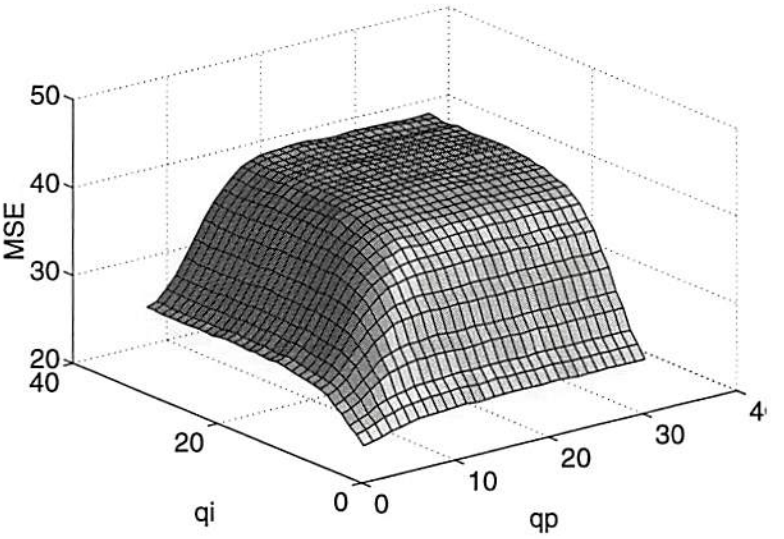
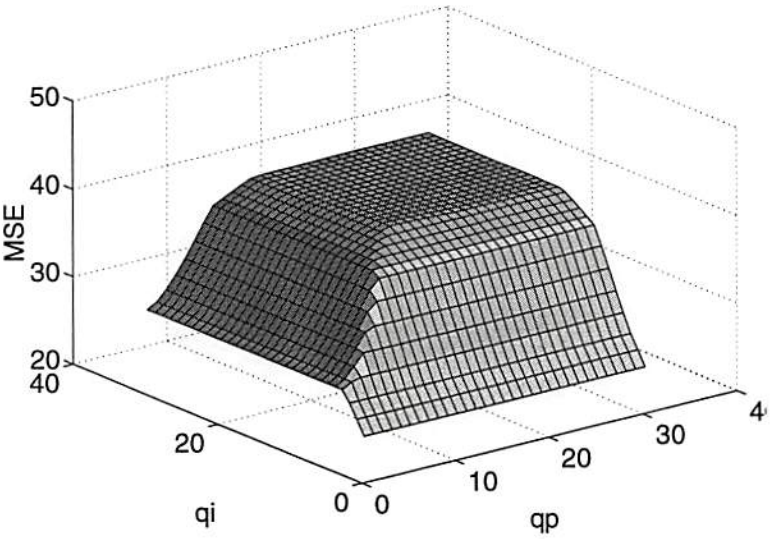


Figure 4.15: The dotted line is the MSE of a P frame in the football sequence, with respect to the MSE of its reference frame. The quantization scale of the P frame, q_P, is fixed at 10. The curve is approximated by a linear-constant function, indicated by solid line. The circles indicate the two control points, at $q_I = 5$ and $q_I = 13$. The corner point is at $q_I = q_P = 10$.



(a) Original Data



(b) Approximated Data

Figure 4.16: (a) Original measured data and (b) reconstructed with B-frame model, of a B frame in the football sequence, as a function of q_I and q_P, with q_B fixed at 10.

4.4 Bit-Rate Control with Approximated R-D

In this section, we apply the approximation model to MPEG video encoding. The gradient-based algorithm in Chapter 3 is used for the rate control, with the R-D for I frames substituted by the approximated data using procedures defined in Section 4.2, and R-D for P and B frames substituted by the approximated data using procedures defined in Section 4.3.

4.4.1 Revised Gradient-Based Algorithm

To evaluate the effectiveness of the proposed model, we apply the approximation model to the gradient-based rate control algorithm introduced in Chapter 3, by substituting the rate and distortion functions with the approximated values. Because of the model error in rate, the original strictly-constant-rate for GOP constraint may no longer be satisfied, but we expect the buffer constraints will still be satisfied most of the time because most of the errors are due to the B frames, which consume the fewest bits. Considering the fact that the model errors in B frames are relatively large, we can further improve the solution by re-allocating bits for the B frames after a solution for the GOP is obtained. This is done by encoding the I and P frames using the solution from the gradient-based algorithm, and calculating the total number of bits remaining for the B frames, which we denote as T_B. Using this available bit budget, the bit allocation for B frames is then re-optimized. The additional optimization procedure does not cost much in terms of computation, because all the reference frames (I and P) are fixed and all the B frames are independent of each other. Denote the rate and distortion functions of B frames as $r_j(q_j)$ and $d_j(q_j)$, where j is the index of B frames. The optimization

problem is to determine quantization scales for B frames ($q_1, q_2, \ldots, q_{N_B}$, where N_B is the total number of B frames in a GOP), such that the overall distortion is minimized, as

$$\text{minimize} \quad \sum_{j=1}^{N_B} d_j(q_j), \qquad \text{subject to} \quad \sum_{j=1}^{N_B} r_j(q_j) \leq T_B. \tag{4.33}$$

The problem can be solved by the method of Lagrange multipliers, using Algorithm 2.1 presented in Section 2.4.1. The function values of $r_j(q_j)$ and $d_j(q_j)$ are obtained by evaluating the function at control points and interpolating by the intra-frame model. By using this approach, the solution is improved and the strictly-constant-rate for GOP are satisfied again.

4.4.2 Experiments and Results

We encode two video sequences, football and table tennis, in CIF format at 1.152 Mbps, using three different configurations: (i) gradient-based method with approximated R-D from the proposed model; (ii) use (i) with additional bit-reallocation for B frames using Lagrange method; (iii) gradient-based method with the original R-D. The GOP was chosen to be size of 6 (IBBPBB) for a reasonable convergence rate (algorithms suitable for higher GOP size will be introduced in the next chapter). The results are shown in Fig. 4.17 and Table 4.5. An additional result from the Software Simulation Group's MPEG-2 encoder [41], which uses an implementation of the TM5 algorithm [40], is also shown for reference. Note that in this experiment, we do not include the adaptive quantization scheme into consideration, while the standard TM5 program we used here includes an adaptive quantization procedure. The computation complexities shown

in the Table are relative to the TM5 algorithm, and are estimated based on the subroutines used in [41], where (i) 13 multiplication operations and 29 addition operations are required for each 8×1 DCT, (ii) two-step search method is used for the motion estimation, in which the spiraling outward full search is applied first for full-pixel displacement, followed by the search for 8 neighboring half-pixel displacement. The motion estimation takes about 90 percent of overall computations in a single-pass encoding. We assume that the memory is large enough to hold all the intermediate data including the motion vectors, reconstructed reference frames, DCT coefficients, etc., so that many of the operations only have to be done once during the evaluation of R-D data on the control points. Note that the relative increase in complexity with respect to TM5 will become larger if a faster motion estimation algorithm is used, since the motion estimation is responsible for the bulk of the complexity in the entire encoding process.

	Football		Table Tennis	
	PSNR	Complexity	PSNR	Complexity
Model R-D	33.13	1.68	32.64	1.71
B-Frame Re-Alloc.	33.17	1.70	32.80	1.73
Original R-D	33.17	8.87	32.74	11.35
Test Model 5	32.43	1.00	31.25	1.00

Table 4.5: Average PSNR and computation complexity with different encoding method. The second row is based on the model R-D with additional bit-re-allocation for B frames. The computation complexity is relative to the Test Model 5 algorithm.

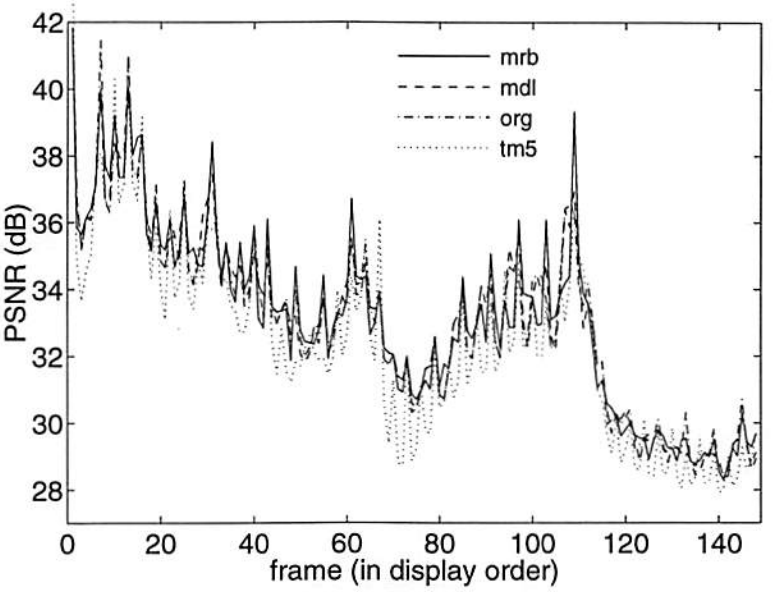
The results show that, by using the approximated model, the number of computations is reduced significantly with very little loss in PSNR. With bit-re-allocating on B frames, we are able to achieve the same PSNR with only a fraction of computation overhead. The results also show that, for the table tennis sequence in Fig. 4.17 (b), the optimum method is capable of adjusting to the scene

changes much faster than the test model 5 algorithm.

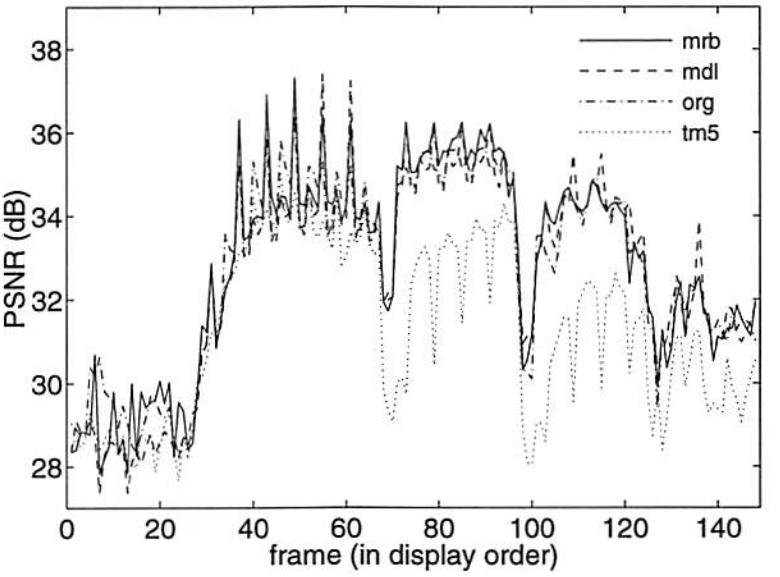
4.5 Conclusions

With the above experiments, we have demonstrated that the proposed model provides a good approximation of rate-distortion characteristic for any given quantization setting. The first application to the gradient-based rate control algorithm shows the same performance can be achieved with only 15 to 20 percent of the computation cost. The model is also applicable to other optimal rate control techniques such as the dynamic programming approach in Section 2.4.3. In addition, our model is also useful for a VBR encoding (e.g. ATM network) scheme to potentially achieve a constant quality.

In the next chapter, we propose a new bit-rate control scheme with relatively low delay and complexity, by using the R-D curve predicted from the previously coded frames.



(a) Football



(b) Table tennis

Figure 4.17: PSNR of image frames for Football and Table tennis. In each figure, *mrb*: gradient-based method using the approximated R-D by the proposed model, with additional bit-re-allocation for B frames; *mdl*: gradient-based method using the approximated R-D only; *org*: gradient-based method using the original measured R-D; *tm5*: Test model 5 algorithm. Note that for the Table Tennis sequence, the TM5 algorithm does not handle the scene change (at 68^{th} frame) well.

Chapter 5

Fast Bit-Rate Control Schemes

5.1 Introduction

In the previous chapter, we have demonstrated that the optimum bit-rate control techniques achieved a better encoding quality, and the spline approximation R-D model reduced the computation complexity without degrading the quality. However, the computational complexity is still higher than those buffer-state-feedback based approaches such as MPEG Test Model 5 (TM5). Even when the computing power is high enough to carry out the bit-rate-control operations in real-time, it still cannot be used for real-time encoding, because the entire GOP has to be fed-in and processed before encoding its first frame, i.e., there is a delay time of one GOP. Therefore, the method is only suited for off-line encoding.

In this chapter, we focus on real-time encoding applications. Currently, many rate-control schemes for real-time encoding are based on schemes similar to the TM5 algorithm, and only control the rate, without a mechanism to monitor the distortion. In order to improve and stabilize the encoding quality, we still follow

the framework of rate-distortion optimization, but only allow one frame delayed for pre-analysis. The new algorithm is an enhancement of TM5, by adding the approximated R-D model in the previous chapter to control the quality. Note however that, one problem in TM5 is that it does not include a mechanism prevent the encoder buffer from overflowing. It only monitors the buffer occupancy and gives warnings whenever overflow occurs. Although the overflow rarely occurs when using the standard decoder buffer size recommended by MPEG, it is not true when the buffer size is smaller due to short delay requirement. In this chapter, we still follow the same rule as for the TM5 algorithm and relax the buffer constraints.

This chapter is organized as follows. In Section 5.2 we present a fast R-D optimized bit-rate control scheme which uses the predicted R-D data to control the quality. In Section 5.3, based on the new fast algorithm and with the help of studies on the human visual system (HVS), we present a new quantization scheme to improve visual quality, and show some MPEG encoding results.

5.2 Fast Bit-Rate Control with Predicted R-D

Consider the fact that, unless there is a scene change, the contents of image frames are usually similar to each other within a short period of time (e.g. within a GOP). When encoding a GOP, it is reasonable to assume that the R-D characteristics of a future, not yet coded, frame are similar to the most recently coded frame of the same type. In this section we propose an algorithm which uses models analogous to those considered earlier but where we now assume that only the current frame will be modeled while we use models based on already encoded frames for the remaining frames in the GOP. Thus, the R-D data of un-coded frames remaining

in the GOP is predicted using their nearest frame of the same type. Because the predicted R-D data is not accurate enough, additional procedures are incorporated to control the rate. The intra-frame spline approximation method introduced in Section 4.2 is still used here for reducing computations.

5.2.1 Control Procedures

In the new control procedure, we use optimization approaches similar to those presented in the previous chapters, and still consider a GOP as a basic unit in the optimization. We follow the same notation as in Section 2.3.2 for TM5, and denote the number of frames for each frame-type within a GOP as N_I, N_P, N_B. The total number of bits allocated for the GOP is then derived by

$$B = B_0 + (N_I + N_P + N_B) \cdot \frac{R}{F}, \tag{5.1}$$

where B_0 is the number of bits left (or over-used if it is negative) from previous GOP, R is channel rate in bits per second, and F is frame rate in frames per second. The encoding procedure is as follows. After a frame of image data arrives, we first measure and approximate its R-D functions, $r(q)$ and $d(q)$. Unlike the algorithms presented in the previous chapters which allow all frames of a GOP to be read and analyzed before any real encoding takes place, we now have to encode the input frame and deliver it to the output stream immediately. In order to complete the optimization procedure, the R-D data of future frames are substituted by the data from most recently coded frame of the same type. For example, the latest P frame model is used for all future P frames remaining in the GOP, and the latest B frame model is used for all future B frames remaining in the GOP. Therefore, we

need to keep three sets of R-D data for the future frames, denoted as $r_I(q)$, $d_I(q)$, $r_P(q)$, $d_P(q)$, $r_B(q)$, $d_B(q)$, for I, P, B frames respectively. With this R-D data, the optimization procedure is operated and a solution is derived. The solution is a set of quantization scales for all frames in the GOP, but only the one for the current input frame is used to encode the current frame. After the current frame is encoded, we count the actual number of bits consumed by the frame, subtract it from B in (5.1), and then remove the current frame from the GOP (so the number of frames is decrease by one). In the next step, we read the next frame and repeat the same optimization procedure using the new value of B and new structure of GOP. The entire procedure is then repeated again until all frames in the GOP are encoded.

Two possible choices of optimization criteria are tested in this dissertation, including minimization of total MSE *(Minimum MSE)*, and minimization of difference in MSE between consecutive frames (*Smooth MSE)*.

Minimum MSE

The first optimization criterion is to minimize the total MSE, as we did in the previous chapters. Because we do not have access to the data from future frames, it is not possible to build an inter-frame model as presented in Section 4.3. To keep the computation simple, we consider the monotonicity property defined in Definition 2.1, Section 2.4.3. If the MPEG coder possesses this property, a better quality in the reference frame (I and P) will lead to a better total coding efficiency, which has been observed in most of our MPEG encoding experiments (and also in [46]). Hence, it is reasonable to confine the operating point within the domain $q_I \leq q_P \leq q_B$. Using the three sets of R-D data, one for each frame type, the

total MSE becomes

$$D(q_I, q_P, q_B) = N_I \cdot d_I(q_I) + N_P \cdot d_P(q_P) + N_B \cdot d_B(q_B), \tag{5.2}$$

and the optimization problem becomes to minimize $D(q_I, q_P, q_B)$ subject to

$$N_I \cdot r_I(q_I) + N_P \cdot r_P(q_P) + N_B \cdot r_B(q_B) \leq B, \tag{5.3}$$

$$q_I \leq q_P \leq q_B, \tag{5.4}$$

where B is total number of bits available for a GOP. Because there are only three independent variables and there is no inter-frame dependency involved, it can be efficiently solved by the Lagrange multiplier method in Algorithm 2.1.

Algorithm 5.1 *Minimum MSE*

Step 1. *Initialize the value of N_I, N_P, N_B and the total bit-budget of a GOP by (5.1).*

Step 2. *Read an image frame (which we call the current frame) and do the necessary motion compensated prediction if it is a P or B frame, then transform the data into DCT domain.*

Step 3. *Evaluate and approximate the rate-quantization $r(q)$ and distortion-quantization $d(q)$ functions for the current frame, using the intra-frame approximation method presented in Section 4.2. If the current frame is an I frame, substitute the results into $r_I(q)$ and $d_I(q)$. If it is a P frame, substitute the results into $r_P(q)$ and $d_P(q)$. Otherwise (must be a B frame), substitute the results into $r_B(q)$ and $d_B(q)$.*

Step 4. *Minimize total MSE in (5.2) subject to the constraints in (5.3) and (5.4). The solution is denoted as* (q_I^*, q_P^*, q_B^*). *Note that at this stage, compared to the TM5 algorithm, we have solved the "bit allocation" and "rate control" in one step.*

Step 5. *If the current frame is an I frame, use* q_I^* *to encode the frame. If the current frame is a P frame, use* q_P^* *to encode the frame. Otherwise (must be a B frame), use* q_B^* *to encode the frame.*

Step 6. *Calculate the actual number of bits consumed by the current frame, and subtract it from* B. *Also, depending on the current frame type (I, P or B), decrease* N_I, N_P, *or* N_B *by one.*

Step 7. *If the current frame is the last frame of GOP, assign* B *to* B_0, *advance to next GOP, and go to Step 1. Otherwise, advance to next frame and go to Step 2.*

Smooth MSE

Another criterion which often leads to a more stable playback quality is to minimize the difference in MSE between consecutive frames, in addition to maximize the quality within the rate constraint. This can be done by a two-step optimization process. The first step is to minimize the MSE difference, by using the following procedures. Based on the current frame type, we pick one variable in $\{q_I, q_P, q_B\}$ as a primary variable. For example, suppose the current frame is an I frame, the primary variable is q_I. Given $q_I = x$, the quantization scales for P frames and B frames (denoted as $y^*(x)$ and $z^*(x)$ respectively) are derived by

minimizing the MSE difference

$$y^*(x) = \arg\min_y [d_P(y) - d_I(x)], \tag{5.5}$$

$$z^*(x) = \arg\min_z [d_B(z) - d_I(x)]. \tag{5.6}$$

As in the Minimum MSE case, we also add a constraint,

$$d_I \leq d_P \leq d_B, \tag{5.7}$$

which makes the quality of the reference to be better then that of predictive frame, which in general gives better performance due to monotonicity property. Then, in the second, the solution for I frame, denoted as q_I^*, is derived by minimizing the difference between the total bits (generated by the model) and the total bit-budget B,

$$|[N_I \cdot r_I(x) + N_P \cdot r_P(y^*(x)) + N_B \cdot r_B(z^*(x))] - B|, \tag{5.8}$$

over all possible x's (or values for q_I).

If the current frame-type is P or B, the solution of q_P^* or q_B^* can be derived by a similar procedure.

Algorithm 5.2 *Smooth MSE*

Step 1. *Initialize the value of N_I, N_P, N_B and the total bit-budget of a GOP by (5.1).*

Step 2. *Read an image frame (named as current frame for later reference) and do the necessary motion compensated prediction if it is a P or B frame, then transform the data into DCT domain.*

Step 3. *Follow the same procedure as in Algorithm 5.1, Step 3 to derive and update R-D data.*

Step 4. *Derive the solution q_I^*, q_P^*, or q_B^* (according to the current frame type) by using the above double-loop optimization procedure.*

Step 5. *Follow the same procedure as in Algorithm 5.1, Step 5 and Step 6.*

Step 6. *If the current frame is the last frame of GOP, assign B to B_0, advance to next GOP, and go to Step 1. Otherwise, advance to next frame and go to Step 2.*

5.2.2 Experimental Results

We encode the six test video sequences by the proposed algorithms. The results are shown in Table 5.1 and 5.2. The PSNR curves of encoded image frames for GOP size 15 are shown in Fig. 5.1, 5.2, and 5.3. The results from the previous chapter (using gradient-search procedure with approximated R-D plus re-optimization on B-frames) are also shown in the Table for comparison. Note that the computation complexity of the new algorithm is similar to that of TM5 with only 8 additional quantization and encoding operations for each frame. Compared to other operations like motion estimation or DCT, the additional overhead is not significant. It also has the potential to be sped-up further by using a parallel hardware implementation.

Bicycle				
	GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff
Gradient & Model	27.05	45.86	n/a	n/a
Prediction & Minimum MSE	26.92	10.80	27.04	10.19
Prediction & Smooth MSE	26.86	8.83	27.01	7.57
Test Model 5	26.37	27.07	26.49	27.11

Cheer				
	GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff
Gradient & Model	26.59	35.45	n/a	n/a
Prediction & Minimum MSE	26.31	10.04	26.53	10.66
Prediction & Smooth MSE	26.19	6.44	26.40	4.96
Test Model 5	25.86	25.05	26.06	25.86

Football				
	GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff
Gradient & Model	33.17	6.25	n/a	n/a
Prediction & Minimum MSE	33.14	5.49	33.18	4.10
Prediction & Smooth MSE	32.98	7.35	33.10	5.19
Test Model 5	32.43	11.29	32.49	11.01

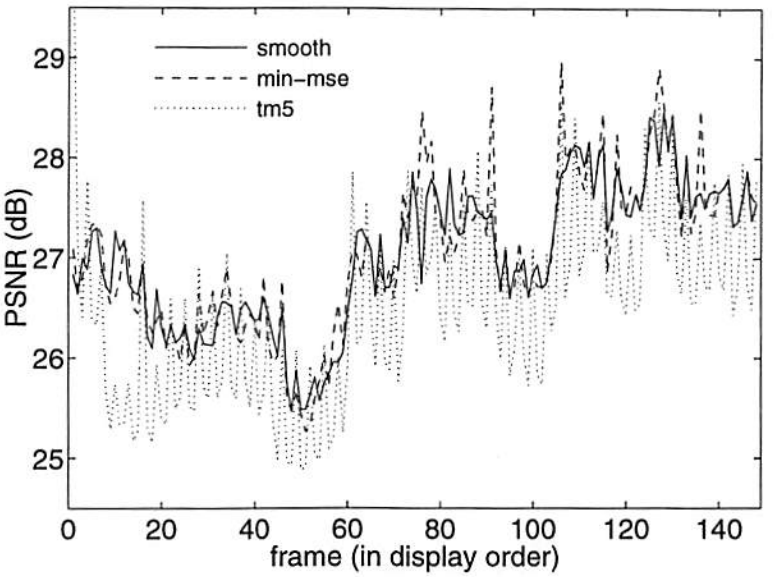
Table 5.1: Average PSNR and first-order difference of MSE for the test sequences. *Gradient & Model*: gradient method with R-D approximated by the model with additional bit-re-allocation for B frames; *Prediction & Minimum MSE*: Predicted R-D with minimum MSE; *Prediction & Smooth MSE*: Predicted R-D with smooth MSE.

Flower				
	GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff
Gradient & Model	27.07	26.25	n/a	n/a
Prediction & Minimum MSE	26.89	15.10	27.62	14.43
Prediction & Smooth MSE	26.63	12.35	27.24	10.35
Test Model 5	25.91	14.04	26.70	15.01

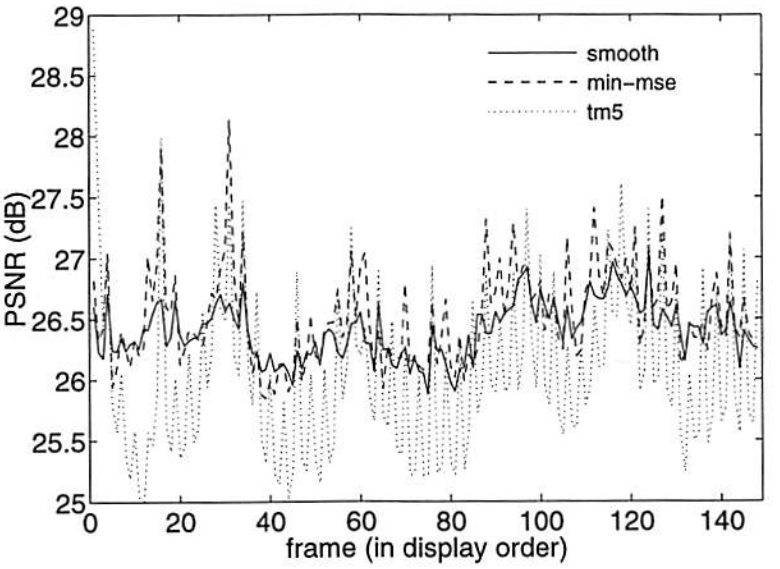
Mobile				
	GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff
Gradient & Model	25.25	30.38	n/a	n/a
Prediction & Minimum MSE	24.70	18.16	25.75	15.19
Prediction & Smooth MSE	24.75	15.58	25.52	10.74
Test Model 5	23.96	30.02	25.07	21.44

Table Tennis				
	GOP 6		GOP 15	
	PSNR	Diff	PSNR	Diff
Gradient & Model	32.80	7.13	n/a	n/a
Prediction & Minimum MSE	32.60	7.91	33.43	7.15
Prediction & Smooth MSE	32.44	6.98	33.14	5.48
Test Model 5	31.25	8.62	32.13	7.58

Table 5.2: (Continued) Average PSNR and first-order difference of MSE for the test sequences. *Gradient & Model*: gradient method with R-D approximated by the model with additional bit-re-allocation for B frames; *Prediction & Minimum MSE*: Predicted R-D with minimum MSE; *Prediction & Smooth MSE*: Predicted R-D with smooth MSE.

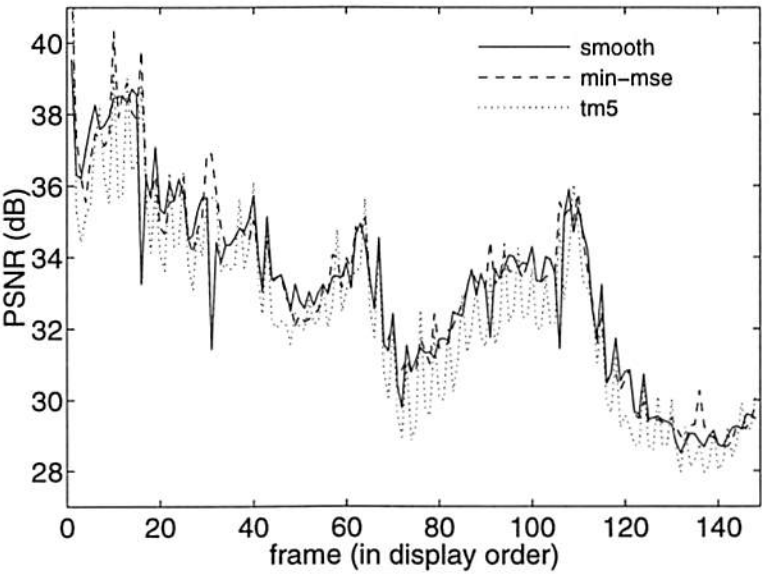


(a) Bicycle

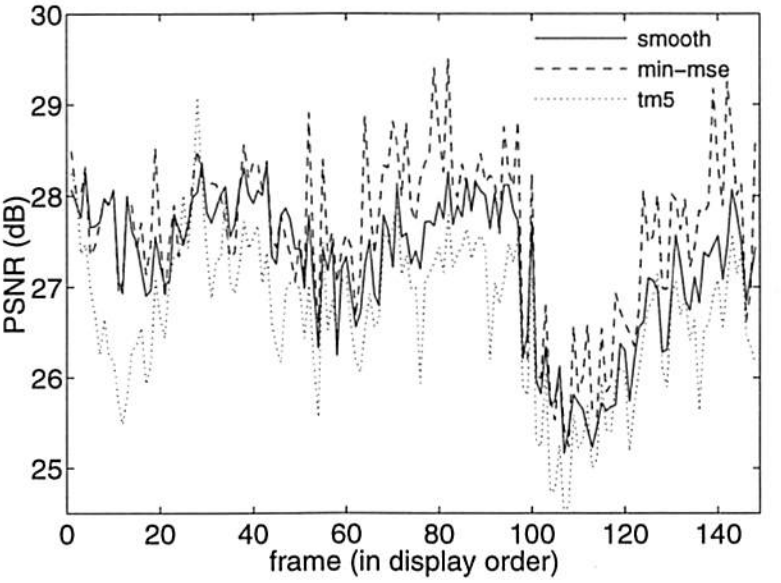


(b) Cheer

Figure 5.1: PSNR of encoded videos (GOP size 15). *smooth*: optimizing by smooth MSE criterion; *min-mse*: optimizing by minimum MSE criterion; *tm5*: Test Model 5 algorithm.

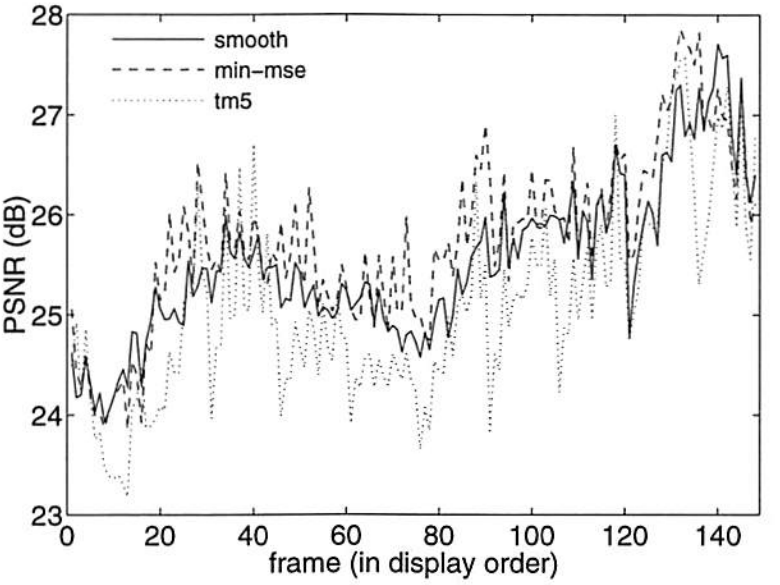


(a) Football

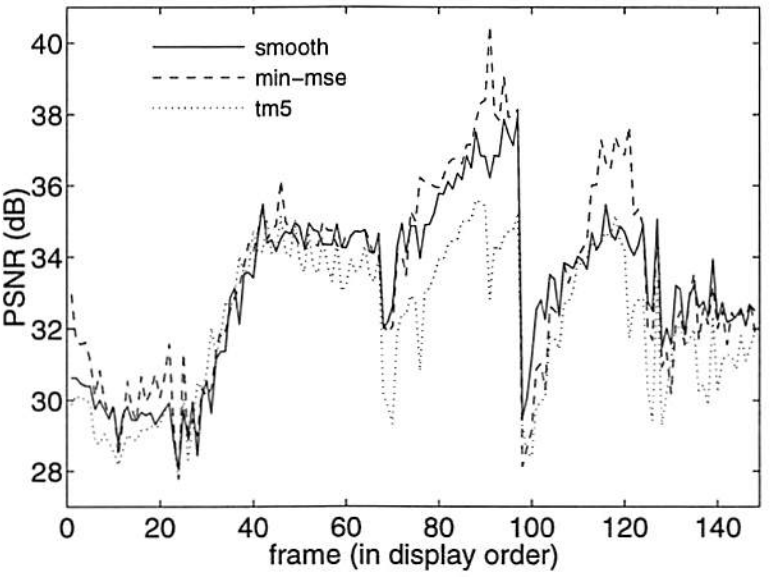


(b) Flower

Figure 5.2: (Continued) PSNR of encoded videos (GOP size 15). *smooth*: optimizing by smooth MSE criterion; *min-mse*: optimizing by minimum MSE criterion; *tm5*: Test Model 5 algorithm.



(a) Mobile



(b) Table Tennis

Figure 5.3: (Continued) PSNR of encoded videos (GOP size 15). *smooth*: optimizing by smooth MSE criterion; *min-mse*: optimizing by minimum MSE criterion; *tm5*: Test Model 5 algorithm. Note that for the Table Tennis sequence, the TM5 algorithm does not handle the scene change (at 68^{th} frame) well.

5.3 Enhancement of Visual Quality

We have proposed several optimization techniques for bit-rate control. Because the cost functions used in the optimization problem are based on the Mean Squared Error (MSE) or Peak Signal to Noise Ratio (PSNR), we observed higher and more stable PSNR in our encoding results. However, it is well known that MSE does not always correspond to quality perceived by the human visual systems. Our techniques are general enough that any quality measure can be used as a cost function, as long as the quality-quantization function is smooth so that the interpolation approximation can be applied. In this section, based on the algorithm proposed in the previous section, we extend the algorithm for improving and stabilizing the visual quality.

5.3.1 Human Visual System

Study of the perception in the Human Visual System (HVS) has been covered within the field of psychophysics and it usually involves intensive experiments and visual tests. Currently, it is still an active research topic. From the study, a model which is particularly useful for image and video compression is the *Multi-Channel Model*, in which an HVS is modeled by a bank of filters, with each filter tuned to a specific band of spatial frequency and orientation [54]. The perception of the signal in each band is governed by two key concepts, the *contrast sensitivity* which accounts for the perception of single band, and the *masking effect* which quantizes the interactions between several different bands. *Contrast sensitivity*, is defined as, for a given spatial frequency, the minimum amplitude of the signal required so that it can be detected. Human eye perceives the output of each band with

different sensitivity. For example, it is more sensitive to lower spatial frequency than to higher frequency, and is more sensitive to horizontal and vertical patterns than to diagonal ones. The second concept, *masking effect*, describe the fact that, if there are more than one signal (in different bands) present in the same area, and the energy of the largest signal is higher than any others by some threshold, all other signals will be masked. There are two types of masking effects for a still image. The first one is due to the DC band of the signal. Because the DC band corresponds to the background luminance, it is also known as *background luminance masking.* On the other hand, if the masking is caused by other higher frequency bands, it is usually called *texture masking.* For video or motion pictures, there is another type of masking, known as *temporal masking*, which is due to the fact that human eyes have to be focused on an object long enough (e.g., 0.1 sec) in order to clearly see the details of that object. Hence, a scene with fast moving objects or frames just after a scene change is usually masked. More details of HVS related studies can be found in [3, 29, 56, 54].

Some of the results from HVS study have been applied to improve the visual quality for the image and video compression. For example, in [9], an algorithm was proposed which utilized the study of the *masking effect* to derive a just-noticeable-distortion (JND) profile. The JND profile was then applied to a wavelet-based image compressor. The same approach is difficult to apply to the MPEG rate control algorithm while maintaining standard decoder compatibility.

For DCT based schemes such as in MPEG, the use of a quantization matrix is a direct application of the study of *contrast sensitivity.* A method for finding a better quantization matrix was proposed in [57]. It might be useful to incorporate the algorithm for bit-rate control, but that would require to allocate additional

bits for different quantization tables from time to time which would add overhead significantly.

In this dissertation, we propose an adaptive quantization scheme which implicitly utilizes the masking effects and can be seamlessly integrated into the rate control algorithm presented in Section 5.2.

5.3.2 Revised Fast Algorithm

In order to minimize the artifacts due to quantization, we first classify the 8×8 DCT block into three categories, namely *flat*, *edge*, and *texture* blocks, and then apply different quantization scales, q_F, q_E, or q_T, to the blocks in each category, as shown in Fig. 5.4. To avoid the blocky artifact in *flat* blocks, the quantization scale q_F should be kept as small as possible. On the contrary, because of masking effects, the quantization scale q_T for texture blocks can be set to a larger value.

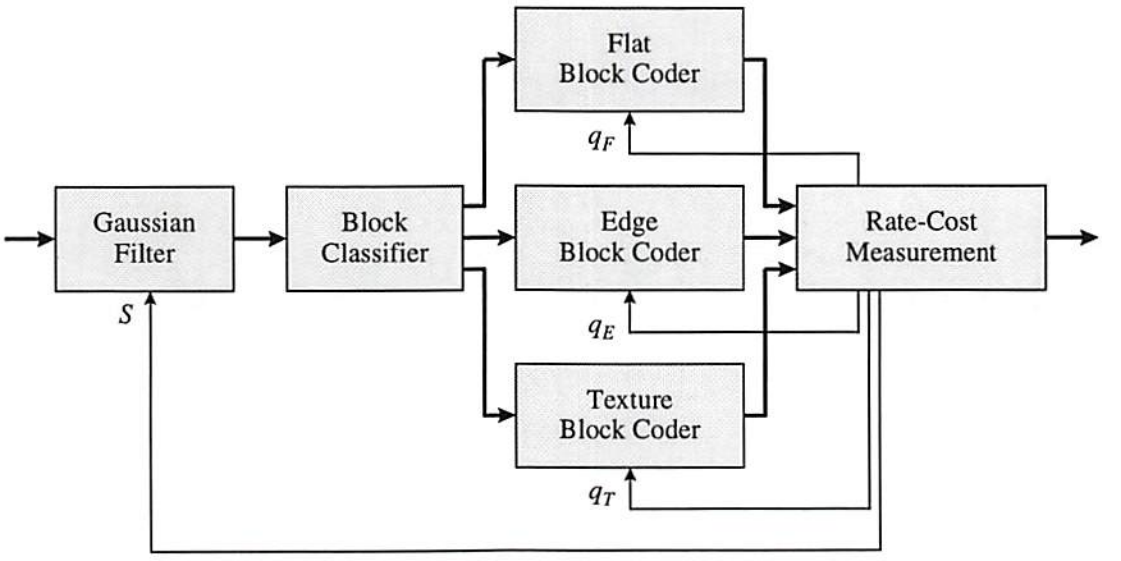


Figure 5.4: Block-classified controller

The following procedures are used to classify the blocks:

Algorithm 5.3 *Block Classification*

- Flat blocks*: The flat block is classified by checking the summation of absolute value of DCT ac coefficients,*

$$\sum_{i=3}^{63} |ac_i| < Threshold, \tag{5.9}$$

where the coefficients are arranged in the zigzag scan order used in JPEG and MPEG. Note that the first two ac coefficients only contribute to smooth variation within a block. Therefore they are not included in the summation.

- Edge blocks*: For those blocks which do not pass the test in (5.9), if at least one of their four nearest neighbors is a flat block, they are considered as edge blocks.*

- Texture blocks*: All the blocks that are not either flat blocks or edge blocks are considered as texture blocks.*

Note that for the classification of blocks in P and B frames, the DCT coefficients of the original image data, not the prediction residue, are used, and thus additional DCT operations are required for these frames. Fig. 5.5 shows classification results from a frame in the Susie sequence, with the $Threshold$ set to 100. In MPEG, the *mquant* is assigned at macroblock level, which consists of four 8×8 DCT block. So, among the q's of four blocks, we choose the smallest one and use it as *mquant* for the encoding.

In Fig. 5.4, a Gaussian-shaped filter is used for pre-filtering. The impulse

response of this filter is defined as

$$h(x,y) = \frac{1}{\sqrt{2\pi}} \cdot exp\left(-\frac{x^2+y^2}{s^2}\right), \tag{5.10}$$

where s determines radius of the filter. The parameter s also determines the trade-off of the rate and distortion and thus can be used as a rate control parameter in a way similar to that of the quantization scales q. The advantage of using s instead of q to control the rate is that the degradation in visual quality as the rate decreases is lesser than that of the case when q is changed. This is due to the fact that the Gaussian function has the same shape but without ripples in both the spatial and frequency domain. Hence, better quality can potentially be achieved by combining s and q in the rate control procedure, especially when bit-rate is low. However, the computational complexity is expected to be higher for our fast control scheme, because we need to repeatedly perform the filtering, encoding and measuring the R-D data for each frame at several different values of s, which requires pre-filtering and DCT for each R-D measurement (in addition to quantization and encoding). For the experiment in this section, to reduce the computation cost, we set s to be a function of channel rate (i.e., set s to a larger value when the channel rate is lower), and fix the value in the entire encoding process.

To incorporate the above block-classification procedure into our fast rate control algorithm, the first choice would be to redefine the cost function as the weighted sum of MSE of different types of blocks, as

$$D(q_F, q_E, q_T) = \sum_i w_F \cdot d_i(q_F) + \sum_j w_E \cdot d_j(q_E) + \sum_k w_T \cdot d_k(q_T), \tag{5.11}$$

where w_F, w_E, w_T are the pre-set weighting coefficients. Now, we have three quantization scale values for each frame, and the measurement of R-D is made independently for each quantization scale, hence the same interpolation R-D approximation scheme can still be applied. In the optimization procedure, there are three q values for each of the I, P, B frames, so now the total number of variables becomes 9. However, we can still independently optimize the cost for each variable, hence the additional computation cost is not particularly high. The parameters which are still left to be determined are the weighting coefficients, w_F, w_E, w_T.

The second method which is more computational efficient is to just keep the ratio of q_F, q_E, q_T constant, as

$$\frac{q_F}{1} = \frac{q_E}{k_E} = \frac{q_T}{k_T}, \tag{5.12}$$

and still use a single parameter, say q_E, as the control parameter for each frame. In this case, the algorithm in Section 5.2 can be applied directly, except that the first optimization criterion (Minimizing the total MSE) is no longer applicable. This is because the MSE measure does not reflect the visual quality, it is meaningless to just minimize the MSE. However, because MSE is still a good indicator for comparing the quality between consecutive frames, the Smooth MSE criterion, which minimize the difference in MSE between consecutive frames, is still useful for achieving stable quality.

Algorithm 5.4 *Smooth MSE with Visual Quality Enhancement*

Step 1. *Based on the specified output bit-rate, determine a value for s in (5.10), and construct a Gaussian filter.*

Step 2. *Initialize the value of N_I, N_P, N_B and the total bit-budget of a GOP by (5.1).*

Step 3. *Read an image frame (which we call the current frame) and apply the Gaussian filter.*

Step 4. *Transform the image into DCT domain (use original image data regardless of the frame type), and do the classification using Algorithm 5.3.*

Step 5. *If the current frame type is P or B, do the motion compensated prediction and transform residue into DCT domain.*

Step 6. *Follow the same procedure as in Algorithm 5.1, Step 3 to derive the R-D data, except now for a given q, three different values of mquant are applied to different block types, with $q_F = q/k_E$, $q_E = q$, and $q_T = q \cdot k_T$.*

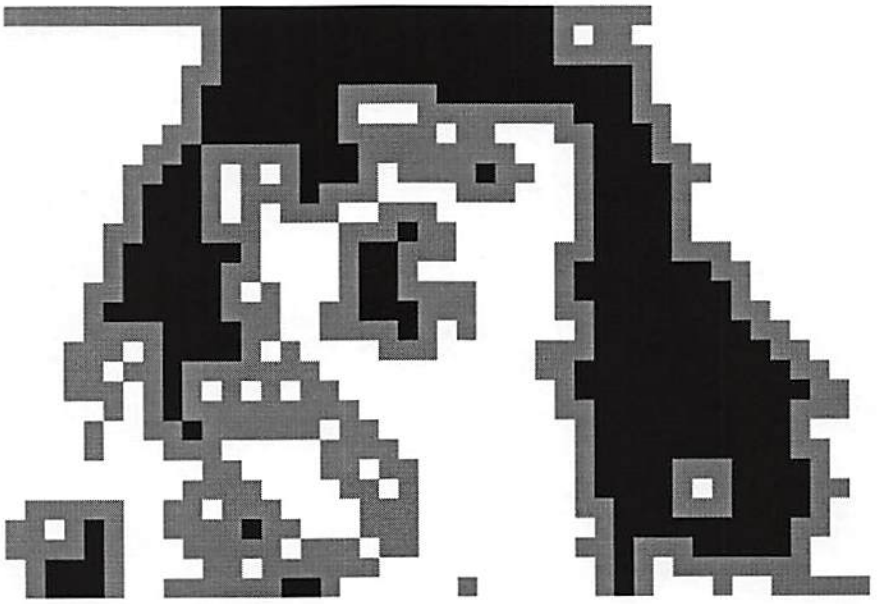
Step 7. *Follow the same procedure as in Algorithm 5.2, Step 4 to derive the solution q_I^*, q_P^*, or q_B^* (according to the current frame type.*

Step 8. *Set q to be q_I^*, q_P^*, or q_B^* (according to current frame type), and set $q_F = q/k_E$, $q_E = q$, and $q_T = q \cdot k_T$, and encode the current frame using q_F, q_E, and q_T for each block type respectively.*

Step 9. *Calculate the actual number of bits consumed by the current frame, and subtract it from B. Also, depending on the current frame type (I, P or B), decrease N_I, N_P, or N_B by one.*

Step 10. *If the current frame is the last frame of GOP, assign B to B_0, advance to next GOP, and go to Step 2. Otherwise, advance to next frame and go to Step 3.*

(a) Original Image



(b) After Block Classification

Figure 5.5: Block classification of a frame in Susie sequence, with *Threshold* set to 100. Gray shade indicates edge blocks and black indicates texture blocks.

5.3.3 Encoding Results

In the experiment, we use 40 frames in the Susie Sequence at 24 frames per second. The Algorithm 5.4 is used for the encoding, with $k_E = 2$ and $k_T = 3$. Two channel rates are chosen to demonstrate the control capability of our algorithm at lower bit-rate, 256 kbits per second and 192 kbits per second. For comparison, we also use the original TM5 algorithm to encode the sequence at the same rates. The file size and location (URL, for Internet web browser) are shown in Table 5.3. The files are in standard MPEG format and can be decoded and playback by any standard MPEG player. Note that the traditional distortion measure, or PSNR, is not included in the data because it does not reflect the visual quality and is not an optimization criterion to our control algorithm. Four reconstructed images of the same frame from the four test results are shown in Fig. 5.6 and Fig. 5.7. In the side by side playback test, we observe that the results generated by TM5 algorithm show clear artifacts in some frames. The temporal variation in quality are also noticeable throughout the playback. The results generated from our algorithm give better and more stable playback quality.

	bit-rate (kbps)	file size (bytes)
Test Model 5	256	53701
Algorithm 5.4	256	53435
Test Model 5	192	40124
Algorithm 5.4	192	40146
File location	http://sipi.usc.edu/~liangjin	

Table 5.3: MPEG files of Susie Sequence, 40 frames, 24 fps.

(a) Test Model 5 Algorithm

(b) Our Algorithm

Figure 5.6: Frame 9 of the Susie sequence, encoded at 256 kbps using (a) the Test Model 5 algorithm and (b) our algorithm. The TM5 algorithm is not quite stable at 256 kbps, while our algorithm still gives a reasonably result.

(a) Test Model 5 Algorithm

(b) Our Algorithm

Figure 5.7: Frame 9 of the Susie sequence. The encoding procedure is the same as that in the previous figure, but encoded at 192 kbps. Again, the TM5 algorithm is not quite stable at 192 kbps, while our algorithm still gives a reasonably result.

5.4 Conclusions

In the first part of this section, we have demonstrated that our proposed fast algorithm which uses predicted R-D characteristics has successfully achieved higher and smoother quality in terms of PSNR with only minor computational overhead. The effects of the two different optimization criteria are also clearly shown in the results. The results show that the performance of the fast algorithm is quite close to the previously proposed one-GOP-delay algorithms.

In the second part, we enhance the algorithm by adding a pre-filtering and block-classification procedure, aimed at better visual quality. The side-by-side playback comparison with the results from TM5 algorithm shows that a more stable and better quality has been achieved by our algorithm. Because our approach is fully decoder compatible, the quality enhancement can be shown by any standard MPEG player.

Chapter 6

Conclusions and Extensions

6.1 Summary of the Research

In this research, our focus has been on bit-rate control techniques to improve the quality of digital video which has to be encoded with a coder with limited buffer size and for a communication channel with limited channel rate. We first surveyed several current bit-rate control algorithms, ranging from light overhead schemes such as MPEG Test Model 5 (TM5) algorithm, to computationally-intensive techniques using Lagrange multiplier and trellis-based dynamic programming methods. For the TM5 algorithm with default parameters, sometimes good quality can be generated, but this is not generally true for any given video sequence. It is also difficult to adjust parameters to improve the quality without the help of manual trial and error adjustments of the encoding. Another problem is, it does not control the buffer to prevent buffer overflow. The R-D optimized control schemes solve all these problems and generate better and more stable results. However,

the costly computation complexity, mainly from the evaluation of rate and distortion functions for many quantization settings, makes the algorithm impractical for many applications, although it is a good benchmark reference for other rate control techniques.

To cope with the high computational complexity problem in the R-D optimized control schemes, we proposed an algorithm based on penalty functions and iterative gradient search. The computational complexity is reduced because we only need to evaluate rate and distortion functions along the search path, which is much less than the requirement for trellis-based approach. Although the algorithm only converges to a local optimum solution, our experiments show that it is close to the global optimum solution.

Our second research topic was on the approximation of rate and distortion functions, which can greatly reduce the complexity of R-D optimized rate control techniques. Previous works were mainly based on specific statistical models. A survey and experiment show that it is difficult to accurately determine the model parameters, and the model error is too large to be useful for R-D optimized rate control algorithm. Therefore, we propose an approximation method based on computing a few R-D points and interpolating the remaining points using spline functions. The inter-frame dependency of R-D functions are also considered and modeled by linear-constant functions. By a complete evaluation on all the quantization settings, we show that the approximation error is small. A direct application to our gradient-based rate control algorithm shows the results are very close to the ones based on the original R-D data, with only about 15% to 20% of computations.

Finally, we proposed a fast algorithm suitable for real-time encoding. The

R-D optimized approach is still used in the algorithm, except that the R-D data for those un-coded frames are predicted from the coded frames. Properties of human visual system (HVS) are also used to enhance the visual quality. The experimental results show that better and more stable quality can be achieved by our algorithm, especially when the channel rate is low (e.g., CIF format at 192 to 256 kbits per second).

All our algorithms and encoding results are compatible with standard MPEG decoders, and thus can be played back by any MPEG decoder with improved quality.

6.2 Future Extensions

Based on the proposed algorithms in this dissertation, some possible extensions of the work are presented in the following.

- **Enhancement of Visual Quality**

  In Section 5.3, we proposed a scheme to enhance the visual quality based on block classification. Additional improvement is possible to make the algorithm more robust to any type of input, including the use of image classification to adaptively adjust the value of k_F, k_T, and s.

- **Rate Control for Scalable Video**

  Scalable video stream defined in MPEG-2 provide a way for a receiver to decode only part of data and get the video at different quality and resolution, depending on the channel condition and player capability. This is useful for video data browsing, HDTV with embedded TV and so on. The rate control algorithm in a scalable video encoder requires to determine the best

policy for assigning bits to different hierarchies, which usually calls for a rate-distortion optimizing procedure. Our R-D approximation model and optimum rate control algorithms can potentially be used in this context to reduce computation complexity.

- **Rate Control for Multiple Program Encoding**

  In the definition of MPEG-2 system stream, several different programs can be multiplexed and transmitted through a single CBR channel. This transmission scheme is now widely used in DirectTV and other digital cable TV system to increase the total number of program channels in a system. Some study has shown that, instead of assigning an equal bandwidth for each program, if an optimization procedure can be performed across different programs, the quality can be improved on all programs. Again, the rate-distortion optimized procedure is still required in this context, and our R-D approximation model and optimum rate control algorithms can potentially be used to reduce computation complexity.

- **Rate Control for Wavelet Video Coder**

  Image compression using wavelets [1] has been demonstrated to have a better performance compared to the DCT-based scheme in terms of both PSNR and visual quality, especially in the low-bit-rate case. Recently, several embedded and multi-rate wavelet encoding schemes were introduced in [49, 52]. The applications to video compression were also presented in [7, 30]. In the embedded coding, the bits are ordered in importance so that the encoder can terminate the encoding at any point to perfectly meet a target rate or distortion. This is particularly suitable for the bit-rate control within an

image (local control). However, when using this scheme for video, it is still necessary to have a strategy in assigning bits to each frame (global control), particularly when the same I, P, B structure as in MPEG is used. Our rate-distortion model and optimization-based rate control scheme is useful in this context.

References

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Proc.*, Vol. 1, pp. 205–220, Apr. 1992.

[2] ATM Forum, *ATM User-Network Interface Specifications, Version 3.0*, Prentice Hall, 1993.

[3] C. A. Burbek and D. H. Kelly, "Spatiotemporal characteristics of visual mechanisms: excitatory-inhibitory model," *Journal of the Optical Society of America*, Vol. 70, pp. 1121–1126, Sep. 1980.

[4] C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. Image Proc.*, Vol. 2, pp. 50–59, Jan. 1993.

[5] J.-J. Chen and H. M. Hang, "A transform video coder source model and its application," in *Proc. of ICIP'94*, vol. II, (Austin, Texas), pp. 962–966, 1994.

[6] J.-J. Chen and D. W. Lin, "Optimal Coding of Video Sequences over ATM Networks," in *Proc. of the 2nd Intl. Conf. on Image Processing, ICIP-95*, vol. I, (Washington, D.C.), pp. 21–24, Oct. 1995.

[7] P.-Y. Cheng and C.-C. J. Kuo, "Scalable video coding using wavelet transform and multiresolution motion representation," in *7th Annual Rockwell conference on Control and Signal Processing*, (Rockwell Science Center, Thousand Oaks, CA), May 1995.

[8] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled Langrange multiplier method," *IEEE Trans. Image Proc.*, Vol. 3, pp. 546–558, Sep. 1994.

[9] C.-H. Chou and Y.-C. Li, "A perceptually tuned subband image coder based on the Just-Noticeable-Distortion profile," *IEEE Trans. Circuits Syst. Video Tech.*, Vol. 5, pp. 467–476, Dec. 1995.

[10] K. W. Chun, K. W. Lim, H. D. Cho, and J. B. Ra, "An adaptive perceptual quantization algorithm for video coding," *IEEE Trans. Consumer Electr.*, Vol. 39, pp. 555–558, Aug. 1993.

[11] T. G. Cover and J. A. Thomas, *Elements of Information Theory*, ch. 13, John Wiley & Sons, Inc., 1991.

[12] G. Dahlquist and A. Bjorck, *Numerical Methods*, Prentice-Hall, 1974.

[13] W. Ding and B. Liu, "Rate-quantization modeling for rate control of MPEG video coding and recording," in *Proc. of IS&T/SPIE Digital Video Compression '95*, (San Jose, CA), pp. 139–150, Feb. 1995.

[14] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. Circuits Syst. Video Tech.*, Vol. 6, pp. 12–20, Feb. 1996.

[15] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, Vol. 11, pp. 399–417, 1963.

[16] R. Fletcher, *Practical Methods of Optimization*, Wiley, 1987.

[17] J. Foley, A. van Dam, S. Feiner, and J. Houghes, *Computer Graphics, Principles and Pratice*, ch. 3, pp. 72–81, Addison Wesley, 1990.

[18] E. Frimout, J. Biemond, and R. L. Lagendijk, "Forward rate control for MPEG recording," in *Proc. of SPIE Visual Communications and Image Processing '93*, (Cambridge, MA), pp. 184–194, Nov. 1993.

[19] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Chapman & Hall, 1997.

[20] C.-Y. Hsu, A. Ortega, and A. Reibman, "Joint Selection of Source and Channel Rate for VBR Video Transmission under ATM Policing Constraints," *IEEE J. on Sel. Areas in Comm.*, 1997. To appear.

[21] *ISO/IEC 11172 (MPEG-1): Coding of moving pictures and associated audio - for storage at up to about 1.5 Mbits/s*, Nov. 1992.

[22] *ISO/IEC 13818 (MPEG-2): Generic coding of moving pictures and associated audio information*, Nov. 1994.

[23] *ITU-T Recommendation H.261: Video codec for audiovisual services at $p \times 64$ kbits*, Mar. 1993.

[24] *DRAFT ITU-T Recommendation H.263: Video coding for low bitrate communication*, July 1995.

[25] J. Katto and M. Ohta, "Mathematical analysis of MPEG compression capability and its application to rate control," in *Proc. of ICIP'95*, vol. II, (Washington, D.C.), pp. 555–559, 1995.

[26] G. Keesman, I. Shah, and R. Klein-Gunnewiek, "Bit-rate control for MPEG encoders," *Signal Processing: Image Communication*, Vol. 6, pp. 545–560, Feb. 1995.

[27] J. Lee and B. W. Dickinson, "Joint optimization of frame type selection and bit allocation for MPEG video encoders," in *Proc. of ICIP'94*, vol. II, (Austin, Texas), pp. 962–966, 1994.

[28] D. LeGall, "MPEG: a video compression standard for multimedia applications," *Communications of the ACM*, Vol. 34, pp. 46–58, Apr. 1991.

[29] G. E. Legge and J. M. Foley, "Contrast masking in human vision," *Journal of the Optical Society of America*, Vol. 70, pp. 1458–1471, Dec. 1980.

[30] J. Li, P.-Y. Cheng, and C.-C. J. Kuo, "A wavelet transform approach to video compression," in *SPIE Symposium on OE/Aerospace Scusing*, (Orlando, FL), Apr. 1995.

[31] D. W. Lin, M.-H. Wang, and J.-J. Chen, "Optimal delayed-coding of video sequences subject to a buffer-size constraint," in *Proc. of SPIE Visual Communications and Image Processing '93*, (Cambridge, MA), pp. 223–234, Nov. 1993.

[32] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "Gradient-based buffer control technique for MPEG," in *Proc. of SPIE Visual Communications and Image Processing '95*, (Taipei, Taiwan), pp. 2502–2513, May 1995.

[33] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "A gradient-based rate control algorithm with applications to MPEG video," in *Proc. of ICIP'95*, vol. III, (Washington, D.C.), pp. 392–395, 1995.

[34] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "Cubic spline approximation of rate and distortion functions for MPEG video," in *Proc. of IS&T/SPIE Digital Video Compression '96*, (San Jose, CA), pp. 169–180, Feb. 1996.

[35] L.-J. Lin, A. Ortega, and C.-C. J. Kuo, "Rate control using spline-interpolated R-D characteristics," in *Proc. of SPIE Visual Communications and Image Processing '96*, (Orlando, FL), pp. 111–122, Mar. 1996.

[36] D. G. Lueberger, *Linear and Nonlinear programming*, Addison-Wesley, 1984.

[37] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performace models of statistical multiplexing in packet video communications," *IEEE Trans. on Comm.*, Vol. 36, pp. 834–843, July 1988.

[38] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, Chapman & Hall, 1997.

[39] MPEG-1 Encoder version 1.3, Berkeley Plateau Research Group <ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/mpeg/mpeg_encode-1.3.tar.Z>.

[40] MPEG-2, *Test Model 5 (TM5) Doc. ISO/IEC JTC1/SC29/WG11/93-225b*, Test Model Editing Committee, Apr. 1993.

[41] MPEG-2 Encoder v. 1.1a, MPEG Software Simulation Group <ftp://ftp.netcom.com/pub/cfogg/mpeg2/mpeg2codec_v1.1.tar.gz>.

[42] A. Ortega and K. Ramchandran, "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to MPEG video coders," in *Proc. of IS&T/SPIE Digital Video Compression '95*, (San Jose, CA), pp. 129–138, Feb. 1995.

[43] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximation," *IEEE Trans. Image Proc.*, Vol. 3, pp. 26–40, Jan. 1994.

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

[45] M. R. Pickering and J. F. Amold, "A perceptually efficient VBR rate control algorithm," *IEEE Trans. Image Proc.*, Vol. 3, pp. 527–532, Sep. 1994.

[46] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Proc.*, Vol. 3, pp. 533–545, Sep. 1994.

[47] A. R. Reibman and B. G. Haskell, "Constraints on Variable Bit-Rate Video for ATM Networks," *IEEE Trans. on CAS for video tech.*, Vol. 2, pp. 361–372, Dec. 1992.

[48] P. Sen, B. Maglaris, N. E. Rikli, and D. Anastassiou, "Models for packet switching of variable-bit-rate video sources," *IEEE J. on Sel. Areas in Comm.*, Vol. 7, pp. 865–869, June 1989.

[49] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Proc.*, Vol. 41, pp. 3445–3462, Dec. 1993.

[50] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoust., Speech, Singal Proc.*, Vol. 36, pp. 1445–1453, Sep. 1988.

[51] *MPEG Video Simulation Model Three, ISO, Coded Representation of Picture and Audio Information*, 1990.

[52] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Proc.*, Vol. 3, pp. 572–588, Sep. 1994.

[53] K. M. Uz, J. M. Shapiro, and M. Czigler, "Optimal bit allocation in the presence of quantizer feedback," in *Proc. of ICASSP'93*, vol. V, (Minneapolis, MN), pp. 385–388, Apr. 1993.

[54] C. J. van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatio-temporal model of the human visual system," in *Proc. of IS&T/SPIE Digital Video Compression '96*, (San Jose, CA), pp. 450–461, Feb. 1996.

[55] W. Verbiest, L. Pinnoo, and B. Veoten, "The impact of the ATM concept on video coding," *IEEE J. on Sel. Areas in Comm.*, Vol. 6, pp. 1623–1632, Dec. 1988.

[56] B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Inc., 1995.

[57] A. B. Watson, "Perceptual optimization of DCT color quantization matrices," in *Proc. of ICIP'94*, (Austin, Texas), 1994.

[58] S.-W. Wu and A. Gersho, "Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV," *IEEE Trans. Circuits and Sys. for Video Tech.*, Vol. 1, pp. 100–112, Mar. 1991.

[59] J. Zdepsky, D. Raychaudhuri, and K. Joseph, "Statistically based buffer control policies for constant rate transmission of compressed digital video," *IEEE Trans. Commmun.*, Vol. 39, pp. 947–957, June 1991.