# Rate Control and Constant Quality Rate Control for MPEG Video Compression and Transcoding

Cheng-Yu Pai

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillments of the Requirement

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

January 2006

*Your file Votre référence*
*ISBN: 978-0-494-16291-0*
*Our file Notre référence*
*ISBN: 978-0-494-16291-0*

# ABSTRACT

## Rate Control and Constant Quality Rate Control for MPEG Video Compression and Transcoding

**Cheng-Yu Pai**

The focus of this thesis is the design of rate-control (RC) algorithms for constant quality (CQ) video encoding and transcoding, where CQ is measured by the variance of quality in PSNR (peak signal-to-noise ratio).

By modeling DCT coefficients as having Laplacian distributions, Laplacian rate/models are developed for MPEG-4 encoding and transcoding. These models accurately estimate the rate and distortion (in PSNR) of MPEG-4 compressed bitstreams. The rate model is applied to a CBR (constant bit rate) encoding algorithm. This algorithm offers a better or similar PSNR as compared to the Q2 [7] algorithm with a lower variation in bitrate. Thus, it outperforms Q2.

These models are then applied to CQ video coding and transcoding. Most CBR control algorithms aim to produce a bitstream that meets a certain bitrate with the highest quality. Due to the non-stationary nature of video sequences, the quality of the compressed sequence changes over time, which is not desirable to end-users. To provide a solution to this problem, six CQ encoding algorithms are proposed: the first two are VBR (variable bit rate) algorithms with a fixed target quality (FTQ), the next two are CBR algorithms with FTQ, and the last two are CBR algorithms with a dynamic target quality (DTQ). Within each group of two, the quality is controlled either at the frame level (using the Laplacian rate/distortion model) or at the macroblock level (using the actual distortions). The Viterbi algorithm is used to reduce the computational complexity of the macroblock-level control. Simulation results indicate that the CQ VBR algorithms can precisely meet the target quality with little or no quality variation. The CQ FTQ algorithms attempt to achieve a given fixed target quality while avoiding buffer overflow. For a reasonable target quality, the proposed algorithms can precisely meet the target PSNR with no variation. Since deciding a reasonable target quality is not trivial, the last two proposed CQ algorithms have DTQ. An extra degree of freedom is introduced to balance the quality variation with the accuracy to the target bitrate and the average

quality. Simulation results indicate that the proposed CQ DTQ algorithms offer a similar or higher PSNR while having lower PSNR variance as compared to Q2 and TM5 (Test Model 5) [8] at similar or lower bitrate. Thus, they outperform Q2 and TM5.

With the success of these algorithms, the CQ DTQ encoding algorithms are extended to MPEG-4 video transcoding (bitrate reduction with requantization). These CQ transcoding algorithms can handle the problems that are uniquely present in transcoders, such as the lack of the original sequence and requantization. Similar to their encoding counterparts, these CQ transcoding algorithms have an extra degree of freedom to balance the quality variation with the accuracy to the target bitrate and the average quality. Simulation results indicate that these algorithms offer lower PSNR variance while having similar/lower average PSNR and bitrate when compared with Q2T and TM5T (transcoding version of Q2 and TM5).

Besides proposing MPEG-4 CQ RC algorithms, an MPEG-2 rate-control algorithm is also developed based on TM5. It aims at improving the subjective quality measured by using Watson's DVQ (digital video quality) metric. When compared with TM5, it provides a better DVQ. However, since Watson's DVQ metric is not a standard way to estimate the subjective quality, PSNR is still used in the rest of the research.

# Acknowledgements

I would like to express my deepest and most sincere gratitude to my supervisor – Dr. William E. Lynch. He has given me clear and helpful guidelines throughout my years of study as a Master's and a Ph.D. student. In addition, I wish to thank him for the great amount of time devoted to me and my work.

I wish to thank the scholarship offered by the National Sciences and Engineering Research Council of Canada (NSERC) Post-Graduate Scholarship (PGS-B), NSERC research grants, and the Canadian Institute of Telecommunication Research (CITR). Their financial support allows me concentrating my time and effort on my research.

I would also like to thank my examining committees for their comments and insights. Their suggestions have helped me to improve the thesis.

Finally, I would like to dedicate this work to my family for their love and supports. I thank you all for your patience and your sacrifices. This work is as much yours as it is mine.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| AC | Alternate Current |
| ATM | Asynchronous Transfer Mode |
| AVC | Advanced Video Coding |
| B-frame | Bidirectionally predicted frame |
| BIFS | BInary Format for Scenes |
| BM | Bit Matching |
| CATV | CAble TeleVision |
| CBR | Constant Bit Rate |
| CD | Coefficient Dropping |
| CIF | Common Intermediate Format |
| CQ | Constant Quality |
| CQP | Constant QP |
| CVBR | Controllable VBR |
| dB | Decibel |
| dbquant | Differential Quantization parameter for B-VOP |
| DC | Direct Current |
| DCT | Discrete Cosine Transform |
| DiffServ | Differentiated Service |
| DPCM | Differential Pulse Code Modulation |
| dquant | Differential Quantization parameter |
| DTQ | Dynamic Target Quality |
| DTTB | Digital Terrestrial Television Broadcasting |
| DVD | Digital Versatile Disk |
| DVQ | Digital Video Quality |
| FD | Frame Dropping |
| FGS | Fine-Grain Scalability |
| FLBM | Frame-Level BM |
| FLBMT | FLBM for Transcoding |
| FLCQ | Frame-Level CQ |
| FLCQT | FLCQ for Transcoding |
| FLQM | Frame-Level QM |
| FLQMT | FLQM for Transcoding |
| FR | Full Reference |
| FTQ | Fixed Target Quality |
| GOP | Group of Picture |
| GOV | Group of VOP |
| HDTV | High-Definition TeleVision |
| HVS | Human Visual System |
| IDCT | Inverse DCT |
| IEEE | Institute of Electrical and Electronic Engineers |
| I-frame | Intra frame |
| ISO | International Organization for Standardization |

| | |
|---|---|
| ITU | International Telecommunication Union |
| JND | Just Noticeable Difference |
| KLT | Karhunen-Loeve Transform |
| LPECBR | Laplacian Parameter Estimation CBR |
| MB | MacroBlock |
| MC | Motion Compensation |
| ME | Motion Estimation |
| MMAX | Minimum-MAXimum |
| MPEG | Motion Picture Expert Group |
| mquant | Macroblock quantization parameter |
| MSE | Mean Squared Error |
| MV | Motion Vector |
| MVBM | Macroblock-Level BM |
| MVBMT | MVBM for Transcoding |
| MVCQ | Macroblock-Level CQ |
| MVCQT | MVCQ for Transcoding |
| MVQM | Macroblock-Level QM |
| MVQMT | MVQM for Transcoding |
| NR | No Reference |
| NTSC | National Television System Committee |
| OQP | Optimum QP |
| PDF | Probability Density Function |
| P-frame | Predicted frame |
| PSNR | Peak Signal-to-Noise Ratio |
| q-factor | Quantization factor |
| QM | Quality Matching |
| QP | Quantization Parameter |
| R/D | Rate/Distortion |
| RC | Rate Control |
| RLC | Run-Length Coding |
| RR | Reduced Reference |
| SAD | Sum of Absolute Difference |
| SDP | Stochastic Dynamic Programming |
| SFGS | Spectral FGS |
| SSE | Sum of Squared Error |
| SVCD | Super VCD |
| TB | Token Bucket |
| TM5 | Test Model 5 |
| VA | Viterbi Algorithm |
| VBR | Variable Bit Rate |
| VCD | Video Compact Disk |
| VLC | Variable Length Coding |
| VLD | Variable Length Decoding |
| VO | Video Object |
| VOD | Video-On-Demand |
| VOP | Video Object Plane |

| | |
|---|---|
| VQEG | Video Quality Expert Group |
| VRML | Virtual Reality Modeling Language |

# Chapter 1

# Introduction

## 1.1. Motivation

Uncompressed video sequences contain a large amount of data requiring a great deal of capacity or bandwidth to store or transmit them. Due to the nature of these sequences, redundancies can be removed by means of video compression. However, to achieve high compression, most video compression standards, such as MPEG (Moving Picture Expert Group)-2 [1] and MPEG-4 [2] standards used in this thesis, use lossy compression. This means that distortion is introduced during compression, i.e. the decompressed video will not be identical to the original. In other words, the compressed video sequences contain distortions that are not present in the original sequences.

Traditionally, the way that video service providers charge customers is by selling bandwidth. For end users, higher bandwidth usually means higher quality video. However, from the user's point of view, quality is more important than bandwidth, since most users perceive high bandwidth as only a means to obtain a higher quality.

As video compression algorithms become more sophisticated, higher video quality can be achieved using the same bandwidth. With this in mind, from the service provider's point of view, selling quality to consumers instead of selling bandwidth is financially wiser: any advance made in compression technology will accrue to service providers rather than to consumers.

In fact, the idea of selling the quality rather than the bandwidth is what has happened in the telephony service. At the beginning of the telephony service, what consumers obtained from the service providers is a fixed bandwidth with toll quality. However, as voice compression technologies improved, service providers required less bandwidth to achieve the same toll quality. Therefore, more connections were made using the existing networks without noticeable change in the quality of service. Consequently, selling quality to consumer meant that improvement in voice compression technology accrued to the service providers. It is expected that a similar dynamic will play out for video service providers. However, to make this happen, it is essential to have an effective way to guarantee to users a fixed constant quality.

In video compression, quality can be controlled by rate-control (RC) algorithms. The purpose of an RC algorithm is to control the compressed bitrate according to some criteria. Most RC algorithms use some kind of rate/distortion models to predict the number of bits and the distortion the compressed bitstream would have. Depending on the variability of the bitrate, RC algorithms can be classified as constant bit rate (CBR) or variable bit rate (VBR) algorithms. For CBR algorithms, buffer constraint is often assumed on a leaky-bucket buffer. The buffer is used to control the maximum number of bits to put in a channel during any time interval. A typical buffer constraint is to prevent buffer overflow and underflow. Buffer overflow should be prevented since it results in loss of information and corrupted bitstreams. As for buffer underflow, it is normally handled by stuffing bits into the channel. Most of the research work in the past had emphasized CBR algorithms, since the CBR constraint facilitates transmission of compressed bitstreams over fixed bandwidth channels.

Traditionally, the main goal of these CBR algorithms is to achieve the highest possible quality while meeting a certain target bitrate. Due to the rate constraint and the non-stationary nature of video sequences, encoders have to spend approximately the same number of bits for both easy-to-code and hard-to-code video sections. This results in a better quality for easy-to-code video sections, and a poorer quality for hard-to-code video sections. Thus, the quality of the compressed sequence changes over time, and cannot be guaranteed. However, quality variation is becoming increasingly important for emerging services, such as the quality-on-demand service. To provide a solution for this problem, the main focus of this thesis is to develop constant quality (CQ) rate-control algorithms.

Several ways to achieve CQ are proposed in this thesis. The first scheme aims to achieve a fixed target quality. This method is useful for the quality-on-demand services where the service providers have to guarantee the video quality. However, since the encoding difficulties change over time, having a fixed target quality means that a VBR channel is required. Unfortunately, the CBR channel is still the mainstream of today's communication networks for which target quality cannot be guaranteed. Thus, the second method to achieve CQ is to have minimum variation in quality while achieving the target quality as well as respecting a buffer constraint. This is a best effort method. This means that the best effort is made to achieve the target quality without violating the buffer constraint. When there is not enough bandwidth, the video quality is reduced. One issue with this method is that it is not trivial to decide a reasonable target quality for a given bandwidth. This is because the encoding difficulty changes over time for the same sequence and for different sequences. As a result, a third method to achieve CQ is to

minimize the quality variation by adjusting the target quality dynamically. All of the above three methods to achieve CQ are investigated in this thesis.

Besides having CQ rate-control algorithms for video encoding, it is also important to have CQ control for video transcoding. Video transcoding is the process of converting video from a higher bitrate to a lower bitrate (bitrate reduction), or converting the compressed video from one format into another. It is required for transmitting pre-encoded video on a channel that has lower bandwidth or is congested. Although there are many ways to achieve bitrate reduction, in this thesis, the term *transcoding* refers to bitrate reduction transcoding by requantization. Since it can be impractical to have a fixed target quality, the third method to achieve CQ (with dynamic target quality) is extended from video encoding to video transcoding.

## 1.2. Contributions of this Thesis

There are three main contributions in this thesis: 1) development of Laplacian rate/distortion models for video encoding, 2) development of Laplacian rate/distortion models for video transcoding, and 3) development of constant quality rate-control algorithms for video encoding and transcoding.

The first contribution of this thesis is the development of Laplacian rate/distortion models for video encoding. As mentioned in the previous section, most rate-control algorithms use some form of rate/distortion (R/D) models to predict the number of bits and distortion for a given quantization step size. The behaviours of these rate/distortion models are governed by some model parameters. In general, these model parameters are computed based on the previously encoded frames, and are used to predict what may

happen for the current (or future) frame. This implicitly requires the video sequences to be temporally stationary. However, this assumption is not always true. To avoid this problem, new rate/distortion models are proposed in this thesis that model DCT (discrete cosine transform) coefficients as having Laplacian distributions. The model parameters (Laplacian parameters) are computed from the current input frame. Hence, these new rate/distortion models no longer require the input to be temporally stationary. As discussed in Chapter 3 , since the computational complexity of the Laplacian R/D models is not high, these models can be used for both temporally stationary and non-stationary sequences.

Based on the principles used in the Laplacian rate/distortion model for video encoding, Laplacian rate/distortion models for video transcoding are also developed. Although they are developed using an approach that is similar to their encoding counterparts, these models need to deal with the unique problems presented in transcoding, such as the lack of information about the original sequence, and the requantization of DCT (discrete cosine transform) coefficients.

The last and the most important contribution of this thesis is the development of an effective framework for constant quality control. In all previously reported CQ algorithms, the quality is controlled indirectly through the target bit allocation. This means that these encoders/transcoders really have no control over quality. If the rate control algorithms call for increasing the quality, more bits are allocated to code the frame. However, the exact quality resulting from this bit budget is unknown for these algorithms. Although this indirect way of controlling the quality works well for some cases, it is not intuitive and cannot be extended to guaranteed quality services where the

actual quality is important. On the other hand, the CQ algorithms proposed in this thesis use a different approach to control the quality – they control the quality directly. This means that instead of having a target *bit* allocation, the proposed algorithm starts with a target *quality* allocation. The bit information only overrides the quality information when buffer overflow threatens. This way to control the quality is more intuitive and seems more effective than the previously reported CQ algorithms.

## 1.3. Figures of Merit

Since constant quality rate control is studied in this thesis, it is important to define a) how *quality* is measured, and b) how *constant quality* is measured.

There are two ways to measure the quality (or distortion) of a video – subjective quality and objective quality. Subjective video quality is usually obtained by having human viewers rate the video. Since these measurements are slow and expensive to perform, subjective quality is impractical to use for video encoding/transcoding applications where real-time feedback is needed. Hence, objective quality metrics are used in most video compression research because they are just algorithms that can be automatically computed in real-time. Objective quality metrics are algorithms that predict how humans perceive the degraded signal. Although there are many objective quality metrics, PSNR (peak signal-to-noise ratio) is used as the quality measure in this thesis. The only exception is in Section 4.2, where Watson's DVQ (Digital Video Quality) [3][4] metric (another objective quality measure) is used as the quality measure.

While the term *CQ* is used in the literature, until recently, there was no clear measure of how *constant* the quality is. Thus, it is not possible to quantify the

performance of these algorithms. In this thesis, *CQ* is measured by the variance of the PSNR over time for the entire sequence. This CQ measure is also used in [5] and [6] (PSNR standard deviation). A good CQ control algorithm should have low PSNR variance (ideally zero) while having the average PSNR meets the target quality (for the fixed target quality case), or having a high average PSNR (for the dynamic target quality case).

## 1.4. Thesis Organization

This thesis is organized as follows. Chapter 2 gives the general background information used by this thesis. It starts by presenting an overview of principles used by MPEG-2 and MPEG-4 video compression standards, such as rate-control, motion compensation, transform coding, and entropy coding. Then, it discusses artifacts introduced in video compression that affect video quality. Different quality metrics are also discussed. Finally, it gives a general overview of topics related to video transcoding, such as transcoder architectures, drift errors, and requantization errors.

In Chapter 3, two sets of rate/distortion models are proposed – one set for MPEG-4 video encoding, and one set for MPEG-4 video transcoding. Each of these methods models the DCT coefficients of the original sequence as having Laplacian distributions. These rate/distortion models are used to predict the number of bits and the distortion presented at the output relative to the original uncompressed sequence. Distortion is measured in mean-squared error (MSE). Some issues that present only in transcoding but not in encoding are also discussed and been dealt with.

Chapter 4 proposes two CBR algorithms that use the traditional goal of maximizing the quality. The first algorithm is designed for MPEG-4, and it uses the Laplacian rate model to set the quantization parameter. Simulation results indicate that this new CBR algorithm meets the target bitrate more accurately and has smaller rate variance than Q2 [7]. The other CBR algorithm is designed for MPEG-2, and it uses Watson's DVQ as the quality metric rather than PSNR. As indicated in the results, although this algorithm has a better performance than TM5 (Test Model 5) [8] for low-motion sequences, it performs about the same as TM5 for high-motion sequences. Also, since Watson's DVQ is a non-standard objective quality metric, PSNR is still used as the measure of quality for the rest of this thesis.

In Chapter 5, a total of six CQ rate-control algorithms for MPEG-4 encoding are proposed. Unlike all previously reported CQ algorithms that control the quality indirectly through the target bit allocation, the proposed CQ algorithms control the quality directly. The first two algorithms are designed for VBR applications with the goal of achieving a fixed target quality. The other four algorithms are designed for CBR applications with different goals: the first two algorithms use a fixed target quality, while the other two adjust the target quality dynamically. Within each group of two CQ algorithms, one of them uses frame-level QP (quantization parameter) control, and the other uses MB (macroblock) level QP control. Simulation results indicate that the proposed CQ algorithms have lower PSNR variance while having better or similar average PSNR at the same bitrate. Note that (close to) zero PSNR variance is observed in some cases.

Chapter 6 extends the principle of CQ control from encoding to transcoding. Two CQ algorithms are proposed in this chapter. These algorithms adjust the target quality

dynamically. Similar to the encoding case, these algorithms use the Laplacian rate/distortion model for transcoding. Simulation results indicate that these algorithms have behaviour that is similar to their encoding counterparts – they can give a more constant quality bitstream with similar average quality.

Finally, Chapter 7 summarizes and concludes this thesis. Some directions for possible future research/enhancements are also given.

# Chapter 2

# Background

Raw video sequences contain a large amount of data, which requires a great deal of storage capacity or channel bandwidth. By using advanced video compression algorithms to remove/reduce redundancy, the size of the raw sequences can be significantly reduced with the expense of degraded quality in the compressed video stream. As an example, the uncompressed bitrate of color NTSC (National Television System Committee) video clip (8 bits/pixel, 480x480 pixels/channel, 3 channels/frame, 30 frames/sec) is about 158 Mbps. With compression, this bitrate can be reduced down to 4 to 8 Mbps, depending on the quality requirement [9].

Several digital video compression standards, such as the MPEG (Moving Picture Expert Group) standards, have been proposed and are widely used nowadays. MPEG, formally known as ISO/IEC JTC 1/SC 29/WG 11 [10], was started in 1988 as a working group within ISO (International Organization for Standardization) with the goal to define standards for digital compression of audio-visual signals. Until now, three projects have been approved by ISO as international standards: MPEG-1 [11], MPEG-2 [1], and MPEG-4 [2].

ITU (International Telecommunication Union) is another standard body that develops video compression standards (known as recommendations). Several standards have been approved until now: H.261 [12], H.263 [13], and H.264 [14]. H.264 is the most recent state-of-the-art compression standard, and it has been adopted as MPEG-4 Part 10

(also known as Advanced Video Coding or AVC). However, this thesis will use the MPEG-4 Visual (Part 2) standard since it was the state-of-the-art compression standard at the beginning of the research work. Hence, this chapter will concentrate on technologies used in the MPEG-1, 2, and 4 (Part 2) standards. Further information about MPEG-4 Part 10 and H.264 standards can be found in [15]. Note that, in this thesis, the term *MPEG-4* always refers to MPEG-4 Visual (Part 2), not AVC (Part 10).

*MPEG-1* (the first project of MPEG) was published as the ISO/IEC 11172 standard in 1993. It is aimed at coding moving pictures and the associated audio for digital storage media with VHS quality at about 1.5 Mbps [16], and is used in video compact disk (VCD) technology. *MPEG-2* (ISO/IEC 13818) was introduced in 1995 as a superset of MPEG-1. It is aimed at generic coding of digital television with higher bit-rates [17], and is used in super VCD (SVCD) and digital versatile disk (DVD) technologies. MPEG-2 is backward compatible with MPEG-1. There is no MPEG-3 standard because it was originally anticipated to address high-definition television (HDTV) applications, but it was found that minor extensions to MPEG-2 would suffice for this higher bit-rate, higher resolution application. Thus, the work on MPEG-3 standard was abandoned. *MPEG-4 Visual* (Part 2) was accepted as ISO/IEC 14496 standard in early 2000. It provides the standardized technological elements enabling the integration of the production, distribution, and content access paradigms for digital television, interactive graphics, and interactive multimedia [18][19]. It is targeted for low bit-rate applications. Popular DivX [20], XviD [21], RealVideo [22], and QuickTime [23] digital video formats are based on MPEG-4 technology.

Currently, there are two new MPEG projects under development: MPEG-7 and MPEG-21. MPEG-7 (ISO/IEC 15938) is also known as "multimedia content description interface". Its goal is to create a standard for describing the multimedia content data that supports some degree of interpretation of the information's meaning, which can be passed onto or accessed by a device or a computer code [24]. Started in May 2000, the latest MPEG project is MPEG-21 (ISO/IEC 18034). It is still at an early developmental stage with the goal to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities [25]. Because MPEG-7 and MPEG-21 are aimed for information management/retrieval rather than compression, this thesis will not discuss technologies used by them.

There are many active research areas related to digital video compression, such as rate-control, error resilience, scalable coding, and parallel encoding/ decoding. The main focus of this thesis is rate-control. Thus, only a brief introduction to these concepts is given. More information can be found in the suggested references.

When the compressed video data is transmitted/stored in a channel/media, it is subject to channel/media errors in the form of bit errors or burst errors. *Error resilience* refers to a strategy of controlling error. It usually involves several steps: error detection and localization, resynchronization, data recovery, and error concealment [26]. More information about error resilience can be found in [26],[27],[28],[29].

*Scalable coding* is implemented in MPEG-2 and MPEG-4. It allows decoding of subsets of the bitstream to generate a lower quality version of the video. Four types of scalability can be performed: data partition, spatial scalability, temporal scalability, and

signal-to-noise ratio (SNR or quality) scalability [9]. More details on scalable coding can be found in [1],[30],[31],[32],[33].

*Parallel processing* is the "information processing that emphasizes the concurrent manipulation of data elements belonging to one or more processes solving a single problem." [34] Because compression is a high complexity operation, it can be profitable to explore the parallelization of compression algorithms. In [35], a real-time software-based MPEG-4 encoder was created using parallel processing technology with 20 workstations. Another attempt at parallel parsing of MPEG video was performed in [36].

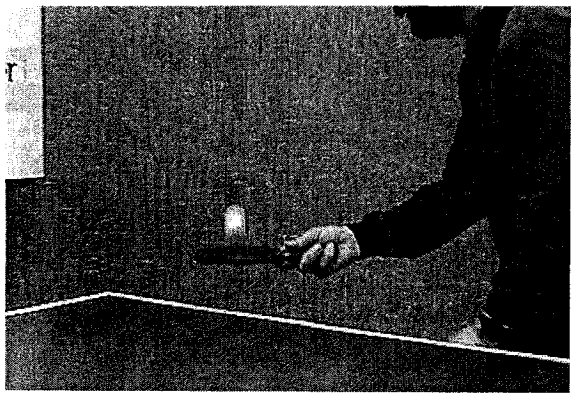
Since video compression standards are built on similar technologies, rather than describing a specific standard, this chapter introduces the general strategies and issues related to video compression. Also, because MPEG-1 and MPEG-2 have been standardized for a while, and they have been widely used nowadays, this chapter concentrates mainly on technologies used in MPEG-1 and 2, which is also used in MPEG-4. In addition, some new technologies introduced in MPEG-4 are also presented.

This chapter is organized as follows: The compression architectures and technologies used in MPEG-1/2/4 are presented in Section 2.1. Because video compression is lossy, i.e., error is introduced in the process of compression, quality issues of compressed sequence are discussed in Section 2.2. Section 2.3 presents concepts related to format (bitrate) conversion (transcoding) used in many current and emerging practical communication applications. Section 2.4 summarizes the chapter.

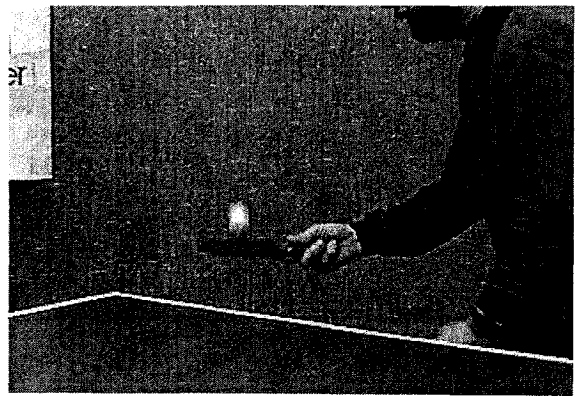
## 2.1. Digital Video Compression Principles

Raw digital video sequences contain a large amount of redundant information that can be exploited for compression. The redundant information can be classified as

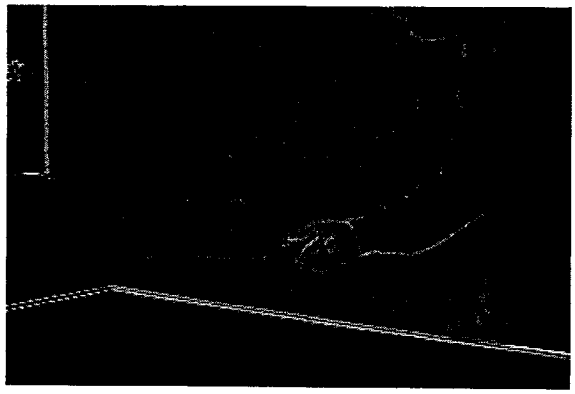
- *Spatial redundancy*: Within each frame, the pixel values are not independent, but correlated with their neighbors.
- *Temporal redundancy*: Since frames are generally sampled at a high rate, 30 frames/sec for instance, the pixel values across frames are also correlated. Consequently, predictive coding can be used. As an example, two consecutive frames (frame 40 and 41) of sequence *table tennis* (30 frame/sec.), and their difference are shown in Figure 2.1.
- *Psycho-visual redundancy*: Due to the characteristics of the human visual system (HVS), humans are more sensitive to low spatial frequencies than high spatial frequencies [30]. For example, the HVS is less sensitive to details near edges or sudden changes. As a result, controlled impairments can be introduced to reduce the bitrate without noticeable degradation of the video quality.



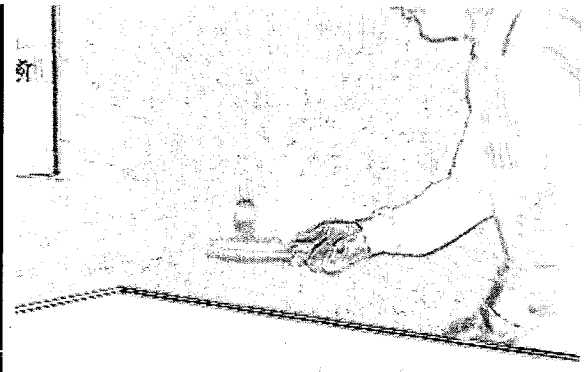
a) Frame 40 of *table tennis*



b) Frame 41 of *table tennis*



c) Difference between frame 40 and 41



d) Inverted difference of c)

Figure 2.1: Temporal redundancy between adjacent frames

This section is arranged as follows: the outlines of the general architecture of an MPEG encoder and a decoder are presented in Section 2.1.1, followed by the input format and definition of compression layers in Section 2.1.2 and 2.1.3. Section 2.1.4 to 2.1.8 present the purpose and functionality of each component used in the encoder.

## 2.1.1. MPEG Encoder/Decoder Architectures

Although there are many MPEG standards, they all use similar technologies to compress the video. The following block diagrams illustrate the encoder (Figure 2.2) and

the decoder (Figure 2.3) architectures of the MPEG-1, 2, and 4 (Part 2) standards. Note that the complexity of the encoder and decoder are not symmetric. The encoder is more complex than the decoder. In fact, the encoder *contains* a decoder, which is used for motion estimation/compensation purposes. The uneven distribution of the complexity allows a simple and cheap decoder to be used at the receiver end of digital video broadcasting.

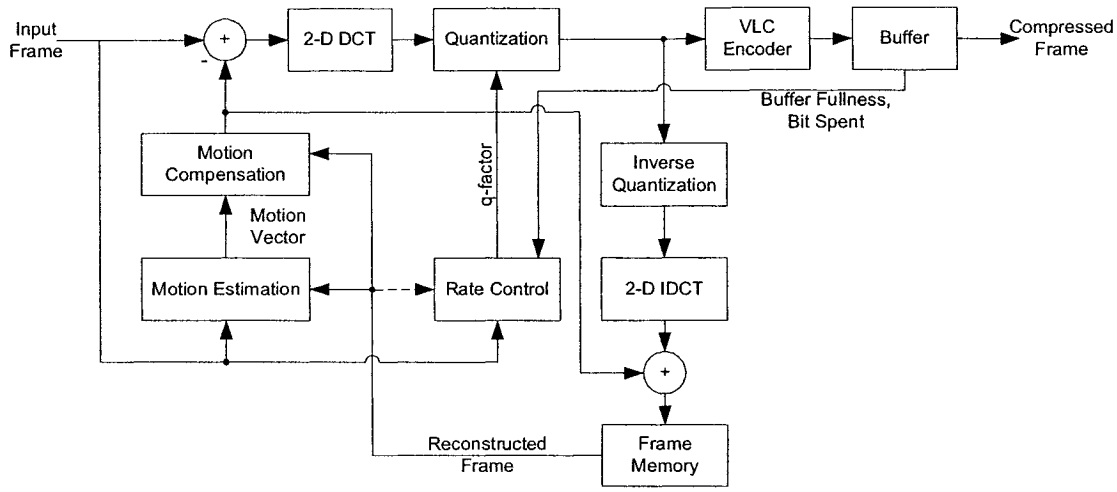



Figure 2.2: MPEG-1/2/4 Encoder block diagram

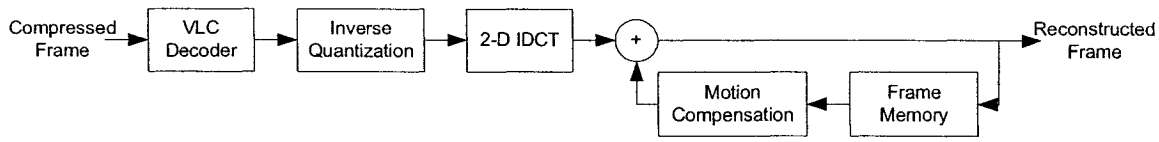



Figure 2.3: MPEG-1/2/4 Decoder block diagram

Note that in the MPEG-4 encoder, the input is not a frame, but a VOP (video object plane). In MPEG-4, the entire video sequence is treated as one or more VOs (video objects). VOP is the representation of the VO at a particular time instance, which is similar to a frame in MPEG-1 and 2. Rather than having a fixed rectangular region (frame), VOPs can have arbitrary shapes. For non-rectangular VOPs, shape information

will also need to be coded to the bitstream. Note that despite the differences, a VOP is coded in a similar fashion as a frame. Thus, in this thesis, the term *frame* is used to refer to a frame (in the MPEG-1/2 case) and a VOP (in the MPEG-4 case).

### 2.1.2. Input Source Format

Scanning is a form of sampling of a continuous two-dimensional signal by representing the image intensity as a one-dimensional waveform. Two scan modes of input frames are supported by the MPEG standards: raster scanning, and interlaced raster scanning. Raster scanning (see Figure 2.4.a) samples each frame once as a picture. Interlaced raster scanning (see Figure 2.4.b) is composed of two interleaved *fields* sampled at different times. It is commonly used in television broadcasting [9].

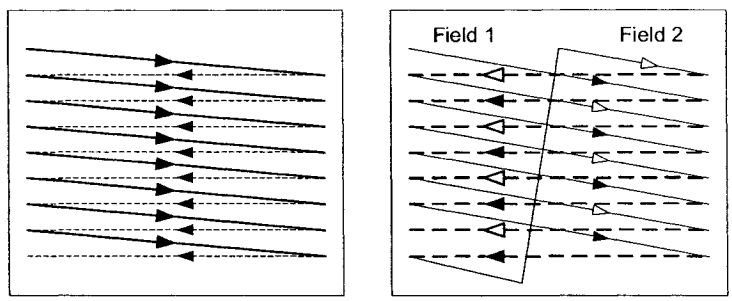



Figure 2.4: Scanning methods of input image [9]

For colored video compression, the color space used in MPEG is YCbCr (luminance/chrominance blue/chrominance red) representation (also known as YUV representation). Since the HVS is less sensitive to small changes in chrominance channels, downsampling of chrominance channels can be used to reduce the bitrate. Three

chrominance channel formats are defined: 4:2:0 format, 4:2:2 format, and 4:4:4 format. The 4:2:0 format is the most commonly used one. It downsamples the chrominance channels by two in both vertical and horizontal directions. The 4:2:2 format downsamples the chrominance channels by two in horizontal direction only. No downsampling is performed in the 4:4:4 format.

### 2.1.3. Layers of Abstraction

MPEG-1/2 video has a hierarchy of layers to help error handling, synchronization, random search, and editing. As shown in Figure 2.5, from the top layer, these layers are: sequence layer, group of picture (GOP) layer, picture layer, slice layer, macroblock (MB) layer, and block layer. The sequence layer defines the entire video sequence, and consists of one or more GOPs. Each GOP consists of one or more pictures. It enables random access of video stream. The picture layer is the primary coding unit, and consists of slices. Slices are used for resynchronization, refresh, and error recovery/concealment. Each slice contains one or more macroblocks in the same line. Each macroblock consists of four luminance blocks and 2, 4, or 8 chrominance blocks depending on the color format used. The macroblock is the basic motion estimation/compensation unit used in MPEG-1/2/4. The block layer is the fundamental layer for spatial compression. Each block contains 8x8 pixels.

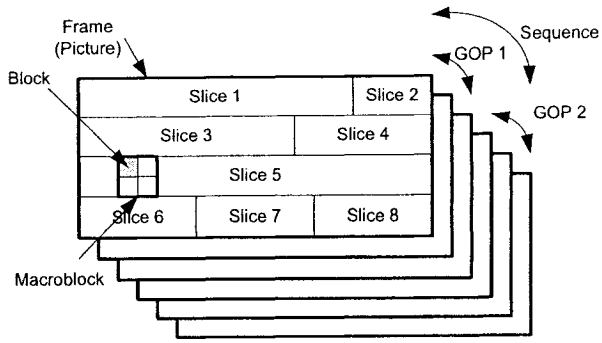


Figure 2.5: Layers of MPEG-1/2 video sequence

There are three types of frames defined in MPEG standards: intra-frame (I-frame), forward predictive-coded frame (P-frame), and bidirectionally predicted frame (B-frame). The P- and B- frames are also known as inter-frames. They are generated by using the motion estimation/compensation algorithm presented in Section 2.1.4. Each GOP contains exactly one I-frame, which is the first frame of the GOP. The GOP may contain P-frames and/or B-frames. I-frames facilitate random access and fast search. P-frames are forward predicted based on the previous I- or P- frame, while the B-frame is predicted based on the previous reference frame (forward prediction) and/or the future reference frame (backward prediction). Only I- and P-frames can be used as a reference frame. Due to the dependencies among I/P/B frames, the encoding order of the frames is different from the display order. For example, the encoding order of a particular GOP structure is shown in Figure 2.6.

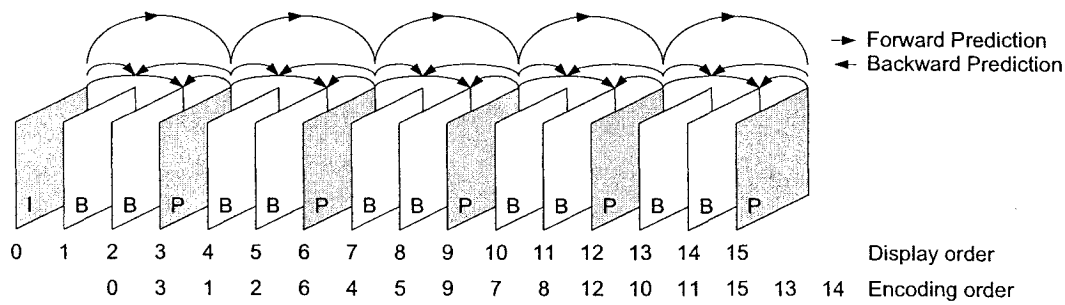


Figure 2.6: Encoding and displaying order of a particular GOP [9]

In general, B-frames can be compressed more than P-frames. Also P-frames can be compressed more than I-frames. Thus, to maximize the compression ratio, one would like to have as many B- (or P-) frames as possible. However, because P-frames propagate error, for resynchronization and error concealment purpose, I- frames should be used once a while. Also, because decoding B-frame requires knowledge of the previous and future reference frames, buffering is needed, which introduces extra delay and may not be suitable for real-time applications.

### 2.1.4. Motion Estimation/Compensation

Motion estimation (ME) and motion compensation (MC) are the key technologies used to reduce temporal redundancy. They use the well-known differential pulse-code-modulation (DPCM) [37] principle. For example, if there is no change in the image intensity between adjacent frames, the temporally most adjacent pixels are the best predictors. Motion estimation is one of the most computation intensive operations in video compression. Its complexity depends on the *best-match* criterion, the range and precision of motion vectors (MVs), and the search algorithm. Depending on the search unit, the ME algorithms can also be classified as block-based ME (used in MPEG-1/2/4) and object-based ME (used in MPEG-4).

The best-match criterion is a cost function indicating the closeness of the reference block with the current block. Two widely used cost functions are *sum of squared error* (SSE) and *sum of absolute difference* (SAD), which are defined as

$$SSE(x,y) = \sum_{i=0}^{N-1}\sum_{i=0}^{N-1}\left(X_{current}[x+i,y+j] - X_{ref}[x+i+dx,y+j+dy]\right)^2 \qquad (1)$$

$$SAD(x,y) = \sum_{i=0}^{N-1}\sum_{i=0}^{N-1}\left|X_{current}[x+i, y+j] - X_{ref}[x+i+dx, y+j+dy]\right| \quad (2)$$

where $X_{ref}$ and $X_{current}$ are the pixel values of the reference frame and the current frame respectively; $x$ and $y$ are the position of the top-left corner position of the block to be matched in the current frame; $dx$ and $dy$ are the MV of the current block; the motion compensation block size is $N$x$N$ pixels. In practice, SAD is more attractive than SSE since it is less computationally intensive (no multiplication is used). Note that in (1), all pixels in the MB are used in the computation. To reduce the computational complexity, some fast ME algorithms may use only a subset of these pixels to estimate the error.

The MVs can have different accuracies for different standards. In MPEG-1/2, MVs have half-pixel accuracy. In MPEG-4, ¼-pixel MV accuracy is allowed. For non-integer MVs, interpolations are required. More accurate motion estimation can be achieved by having smaller MV pixel accuracy with the higher computational complexity. Similarly, the ME can decide the size of the MV (i.e., the range of $dx$ and $dy$). Larger search range can better capture high motion scenes, but with the expense of more computation. A typical search range used in MPEG-2 is ±16 pixels.

There are many ways to perform the ME. The most accurate and the most computationally intensive method is the *full-search* algorithm, which searches all possible candidates within the search range. For ±16-pixel search range with integer MV accuracy, this method would yield $(2\text{x}16+1)^2$ =1089 candidates. Although its computational complexity makes full-search impractical for real-time applications, the MV it generates is the most accurate one, and it is often used as the benchmark for other sub-optimum ME algorithms. To reduce the complexity, many sub-optimum fast-search ME algorithms have been proposed. These fast algorithms reduce the computational

complexity by reducing the number of search candidates while sacrificing the accuracy of the MVs. Some popular fast ME algorithms are three-step search, logarithmic search, hierarchical search, and nearest neighbor search. More details about ME algorithms can be found in [15].

Block-based ME (see Figure 2.7) uses MB to perform ME. It is used in MPEG-1 and MPEG-2 standards. It tries to find the *best match* of the current MB in the reference frame(s), the relative position (motion vector) of the reference MB, and the difference (residue) between them (differential coding). Block-based ME algorithms assume translational movement along a plane parallel to the camera, and that there is no intensity changes between frames. Many block-matching algorithms have been proposed in the past [38],[39],[40]. The full-search algorithm searches the entire reference frame for the best match, and is the most accurate (based on the best-match criterion) but also the most computationally intensive search algorithm. To reduce the complexity, many sub-optimum search algorithms have been proposed in the past. These algorithms reduce the complexities by either reducing the number of candidates to be tested (limiting the search range, for example), or reducing the number of pixels to be checked in the *best-match* criterion, or a combination of both. As an example, by assuming that the frames are sampled at a high rate, and the velocity of the motion is not large, one can reduce the complexity by limiting the search range to $2d+1$ by $2d+1$ as shown Figure 2.7.

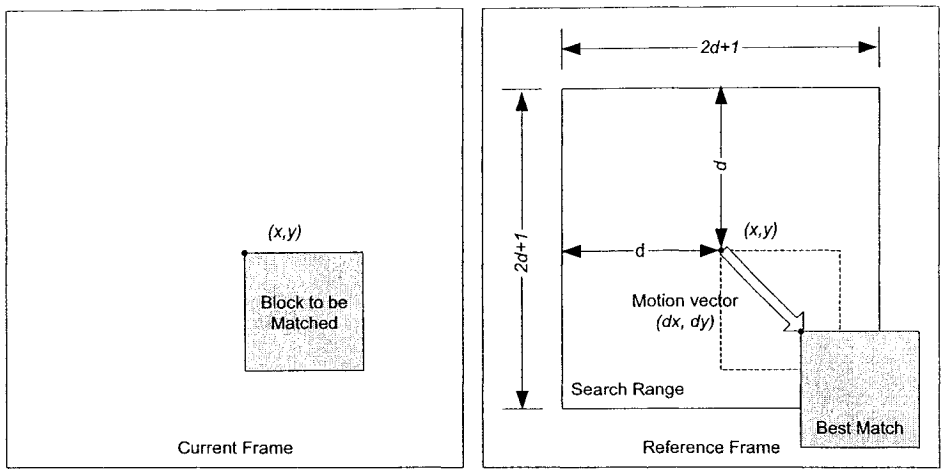



Figure 2.7: Block-based motion estimation with limited search range

There are several issues with block-based ME that may degrade the prediction accuracy, such as

- Real-world objects are not usually square in nature.
- The objects inside the same MB may move in different directions, may rotate, and may move closer or away from the camera.
- Part of the object may appear/disappear (occlusion).

Some of these issues are addressed by using object-based motion estimation. Instead of using the MB as the searching unit, object-based ME first classifies the scene into a collection of VOs of arbitrary shapes and/or textures, and then estimates the movements of the objects.

In MPEG-4, the object functionalities and structures are described by BIFS (binary format for scenes) [41], which are built on several concepts from VRML (virtual reality modeling language). The process of breaking a scene into different objects is called *segmentation*, which is an active research area. Although this task is easy to be performed by humans, video segmentation presents a significant challenge to computers.

It involves the full algorithmic spectrum, extending from low-level image processing to computer vision [42]. The segmentation algorithms can be done automatically by computers [43][44], or interactively by computers with human interventions [45][46]. The underlying algorithm can be region-based [47], contour-based [42][43], or mesh-based [48][49]. *Region-based* methods find regions in the image that are homogeneous with respect to one or more features, and define the region(s) with a segmentation mask. *Contour-based* methods represent regions by their boundaries. *Mesh-based* methods partition the scene into time-varying mesh. The movements of the objects are controlled by the deformation of certain nodes. More details about the object-based coding can be found in [50][51].

### 2.1.5. Transform Coding

While the temporal redundancies are exploited by MC/ME, the spatial redundancies are reduced by transform coding [37][52]. Spatial redundancy exists because values of adjacent pixels are usually highly correlated. Transform coding is used to reduce the correlation in the data by compacting energy into fewer transform coefficients [53]. It transforms signals from the spatial domain to the frequency domain.

The optimum transform for compression in the mean-squared sense is the Karhunen-Loeve transform (KLT) [37]. Because KLT is image dependent and has relatively high complexity, discrete cosine transform (DCT) is used in MPEG standards. DCT [52],[54] is an orthogonal transform that offers performance close to KLT with the advantage of being independent of the images to be transformed. The two-dimensional *N*x*N* DCT and its inverse (inverse DCT or IDCT) are defined as

$$X(u,v) = \frac{2}{N} C(u)C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i,j) \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N} \quad (3)$$

$$x(i,j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)X(u,v) \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N} \quad (4)$$

where $C(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 0 & k \neq 0 \end{cases}$, $u$ ($v$) = 0, 1,..., $N$-1, $X$ and $x$ contain the transformed coefficients and data values respectively. $N$ is the block size. In MPEG-1, 2 and 4 (Part 2), the block size is 8x8 pixels, i.e., $N$=8. By convention, $X(0,0)$ is referred to as DC (direct current) coefficient, and other coefficients are referred to as AC (alternate current) coefficients.

The 2-D DCT and IDCT are separable transforms, i.e., they can be computed by first computing the 1-D DCT/IDCT on rows, then on columns. Fast algorithms exist [55],[56],[57] to bring the computational complexity of 2-D DCT/IDCT from $O(N^2)$ down to $O(N \log N)$. Note that the MPEG standards (MPEG-1, 2, and 4 (Part 2)) do not specify the precision requirement for DCT since it is followed by heavy quantization. However, the precision of IDCT, which is presented in both the encoder and the decoder, should be compliant to the IEEE 1180-1990 standard [58].

Besides using DCT as the way to de-correlate input data, wavelet is another method that could be used. Since it is not used by MPEG standards, no detailed discussion will be given. A comparison between DCT-based and wavelet-based image/video compression is presented in [59]. For still image compression, the study shows that wavelet transform outperforms DCT transform by about 1 dB (decibel) in terms of peak signal-to-noise ratio (PSNR). However, for video coding, wavelet coding yields similar performance to the DCT coding because motion-compensated residues are

less spatially correlated. More information about wavelet-based video coding can be found in [59],[60],[61].

### 2.1.6. Quantization

After transforming the frame from the spatial domain to the frequency domain, quantization is performed on the DCT coefficients (see Figure 2.2). Quantization is required because the DCT only de-correlates the data, and does not reduce the number of bits required to store the data. On the contrary, it even increases the number of bits required to store the data because the input values of DCT are either 8-bit unsigned integers (pixel value) or 9-bit signed integers (residue after motion compensation). However, after the DCT, the coefficients are no longer integers, but real numbers. Even with rounding, the integer part of the coefficients can be up to 12 bits [58]. To reduce the dynamic range of the DCT coefficients, uniform quantization is performed. Note that although quantization compresses the data, it introduces error and degrades the video quality.

In MPEG standards, quantization is performed at the block layer. Each DCT coefficients may be quantized with different step sizes. The actual quantization step size is the product of the quantization parameter (QP) and the corresponding step size in the quantization matrix. As a control mechanism for rate-control algorithm, MPEG allows using different QPs for different macroblocks. However, the same QP is used for all blocks within a macroblock. QP is also known as *q-factor* (quantization factor) or *mquant* (macroblock quantization parameter). The quantization matrix stores values that represent

the relative the importance of each DCT frequency. A larger quantization step size (heavier quantization) results in more compression at the expense of quality degradation.

The quantization matrix is used as a method to emphasize/de-emphasize the importance of DCT coefficients at different frequencies. Because the human visual system reacts like a low-pass filter, low spatial frequency coefficients are more important than high frequency components. This suggests that the quantization step size for high frequencies can be larger than the low frequencies.

Note that the quantization matrices for intra frame and inter frames are normally different. As an example, in the MPEG-2 standard [1], two quantization matrices are defined, which are derived from psycho-visual experiments [62]. Different quantization matrices are used because the DCT input properties of intra and inter frames are quite different. The intra DCT coefficients are transformed from highly correlated pixels. Thus, to take advantage of the HVS, the high frequency components in the intra-frame quantization matrix are usually very large. On the other hand, for inter frames, DCT is used to transform the residue pixels after motion compensation. Since the correlation among adjacent residues is not very strong, there is less emphasis on the high frequency components in the inter-frame quantization matrix.

Besides using the predefined quantization matrices, user-defined quantization matrices are also allowed in the MPEG standards. However, the non-standard quantization matrices must be transmitted to the decoder in order to properly dequantize the DCT coefficients. This increases the bitrate.

### 2.1.7. Entropy Coding

Entropy coding is the last step in the MPEG compression standards. It consists of two lossless compression algorithms: run-length coding (RLC) [37] and variable-length coding (VLC) [37].

The first part of VLC coding is the RLC, which works on a one-dimensional array. However, since DCT coefficients are stored in an 8x8 two-dimensional matrix, they need to be converted into a one-dimensional array. This conversion process is known as *scan*.

Because the DCT coefficients are quantized quite heavily for high frequency components, the high frequency components (lower-right corner of the 8x8 block) are more likely to be zeros, and the low frequency components (upper-left corner) are more likely to be non-zero. Based on this tendency, two scanning methods are provided in the MPEG standards: the zigzag scan (for raster-scanned input), and the alternate scan (for interlaced input). As shown in Figure 2.8, since non-zero coefficients are more likely to happen at the upper-left corner, these scanning methods tend to group the non-zero coefficients near the beginning of the array, and zero coefficients near the end of the array. RLC is used to encode the array by representing the coefficients with a sequence of symbols, each represent a run/level combination. *Run* is the number of zeros before the non-zero coefficient, and *level* is the absolute value of the non-zero coefficient to be coded. These run/level combinations are then mapped to variable-length codewords.

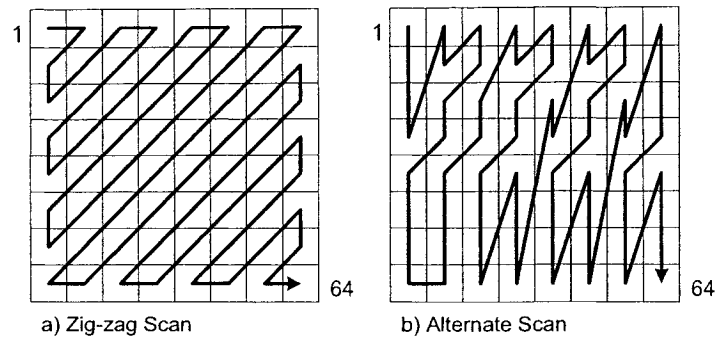



Figure 2.8: Two DCT coefficients scanning method in MPEG standards [9]

VLC uses variable-length codewords for different symbols. It uses shorter codewords for symbols that occur more frequently, and longer codewords for symbols that occur less frequently. From Shannon's theory [63], the average codeword length ($L_{avg}$) is always less than or equal to the entropy ($H$), and is defined as

$$L_{avg} = \sum_k L_k P_k \geq -\sum_k P_k \log P_k = H \tag{5}$$

where $L_k$ and $P_k$ are the length and the probability of the codeword respectively.

## 2.1.8. Rate Control

The rate-control algorithm is probably one of the most important components in the MPEG encoder, and is the main focus of this thesis. It attempts to minimize the distortion while constraining the overall bit budget by adjusting the quantization step sizes. The rate-control algorithm is especially important if the compressed video is to be transmitted over a band-limited communication channel.

Rate-control algorithms can be classified into two categories: constant bit rate (CBR), and variable bit rate (VBR). Generally, CBR algorithms allocate the same number of bits for a fixed time interval (a GOP for example). On the other hand, VBR

algorithms allocate more bits for harder-to-code frames, and fewer bits for easier-to-code frames. As a result, the quality tends to be more constant in VBR than in CBR.

For non-real-time applications, one can even achieve better compression by using multi-pass VBR instead of single pass VBR/CBR. In single pass CBR/VBR, the encoder makes the bit-allocation decisions based only on the knowledge of the previous frames that have been encoded. On the other hand, in the first pass of a multi-pass encoder, the encoder analyzes the video stream to be encoded, and records information about each frame, such as scene changes and rate/distortion statistics. Based on this information, the encoder can distribute more bits for harder-to-code frames (high-motion scenes, for example), and fewer bits for easy-to-code frames (low-motion scenes, for example). The more passes, the more refined the bit-rate distribution will be, i.e., more constant quality for the entire sequence [64].

MPEG did not standardize a rate control algorithm in its standards, and left the decision to the designers. Many rate-control algorithms have been proposed in the past. One of the well-known algorithms in MPEG-2 is TM5 (Test Model 5) [8]. TM5 is recommended, but not required, by MPEG-2 standard as the rate-control algorithm. It works in three steps: target-bit allocation, rate control, and adaptive quantization. The *target-bit allocation* estimates the number of bits available to code the next picture based on the global-complexity measure, which is the product of the bits spent and the average *mquant* of the previously encoded frame of the same type. The *rate control* sets the reference value of the quantization parameter for each macroblock based on the current virtual buffer fullness. The *adaptive quantization* modulates the reference value according to spatial activities inside the macroblock, and derives the value of the

quantization parameters (*mquant*) that are used to quantize the macroblock. Further information about Test Model 5 can be found in [8] and Section 4.2.1.

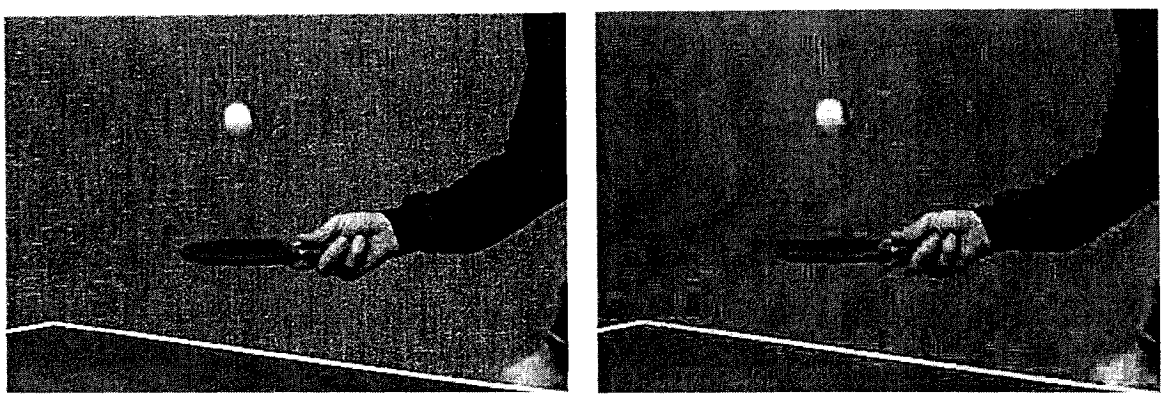
## 2.2. Video Quality Measures

There are two types of compression: lossless compression, and lossy compression. Lossless compression typically has a moderate to low compression ratio, and gives perfect reconstruction from the compressed data without altering the content after decompression. On the other hand, lossy compression can achieve a much higher compression by introducing controlled error. MPEG video compression standards are lossy compressions since quantization is used. In Section 2.2.1, common sources and types of degradations in compressed video are identified. In Section 2.2.2, subjective quality metrics are compared with objective quality metrics in terms of accuracy and cost. These metrics are used to measure the amount of degradation. To bridge the gap between subjective and objective quality metrics, Watson's objective DVQ (digital video quality) metric is also briefly introduced in Section 2.2.3.

### 2.2.1. Factors Affecting Video Quality

There are two components that make video compression lossy: rounding/clipping in the uncompressed domain, and quantization of DCT coefficients. As mentioned in Section 2.1.6, most errors come from quantization. These errors cause degradations in video quality. Several common impairments may occur in the compressed video:

- *Blocking:* Blocking is an artifact of block-based DCT followed by heavy quantization (see Figure 2.9). It occurs due to the discontinuity between adjacent blocks after decompression, especially in low detailed area of image [52]. This effect may also be propagated by motion compensation if the reference picture is already blocky.



a) Original picture b) Blocking effect in the background

Figure 2.9: Blocking effect and edge busyness

- *Blurring/Smearing*: Blurring/Smearing is characterized by the loss of sharpness, especially for high detailed regions. It may be caused by having all (or many) high DCT components being zeros due to heavy quantization.
- *Edge busyness*: It is characterized by spatial distortions near the edge of the object where sharp transition occurs (see the edge of hand in Figure 2.9.b)
- *Mosquito noise*: It is caused by random artifacts along the edges. It may be caused by different edge busyness effects in consecutive frames.

### 2.2.2. Video Quality Metrics

Video quality metrics are used to rate the perceivable degradation. They can be used as measures of the performance of a video encoder. There are two types of video quality: subjective video quality and objective video quality.

Subjective video quality is usually obtained by having human viewers rate the video. However, these measurements are slow and expensive to perform. Real-time feedback is in many cases impossible.

Objective quality metrics are currently most commonly used in video compression. These matrices are algorithms that predict how humans perceive the degraded signal. Although there were many object quality metrics proposed in the past, they can be classified as *statistical quality measures* and *subjective quality predictors*. The statistical quality measures treat the degraded signal as raw data points (numbers); and the quality is just a statistic of the error signal. The subjective quality predictor treats the degraded signal as video signal. In video compression, the degraded (compressed) video signals have unique characteristics that can be easily observed by human viewers. The subjective quality predictors use this knowledge to predict how humans perceive the video.

Two statistical quality measures commonly used today are the mean-squared error (MSE), and the PSNR. They are defined as

$$MSE = \frac{1}{W \cdot H} \sum_{w=1}^{W} \sum_{h=1}^{H} \left( test[w,h] - ref[w,h] \right)^2 \tag{6}$$

$$PSNR = 10 \log \frac{255^2}{MSE} \tag{7}$$

where $W$ and $H$ are the width and height of the frame, and *test* and *ref* are the reconstructed frame and the original frame. Note that 255 is used in (7) because it is the highest possible pixel value of an 8-bit unsigned input. Higher PSNR or lower MSE suggests a better quality.

### 2.2.3. Watson's DVQ

Despite the simplicity of MSE/PSNR, in many situations, they do not reflect the actual perceptual quality correctly. To have a better subjective quality predictor, Video Quality Expert Group (VQEG) [65] was formed in 1997. VQEG is a group of experts working in the field of video quality assessment. The intention of VQEG is to propose an algorithm that gives (or estimates) the subjective quality. Ten algorithms were proposed in the Phase I report [66]. These algorithms estimate the subjective quality by either using the characteristics of the HVS ([3] and [67] for instance), or by measuring the artifacts (such as blocking and ringing) introduced by compression algorithm ([68] for instance). Watson's Digital Video Quality (DVQ) [3][4] metric is one of the algorithms based on the HVS characteristics.

Watson's DVQ (see Figure 2.10) incorporates a simplified HVS model based on sensitivities to spatial, temporal, and chromatic visual signals. It is a full-reference (FR) method, i.e., it needs both the original and the degraded sequences. The output of DVQ is a number that indicates the degree of degradation of the test sequence relative to the reference sequence. The DVQ of the original (uncorrupted) sequence is zero. Higher DVQ means worse quality.

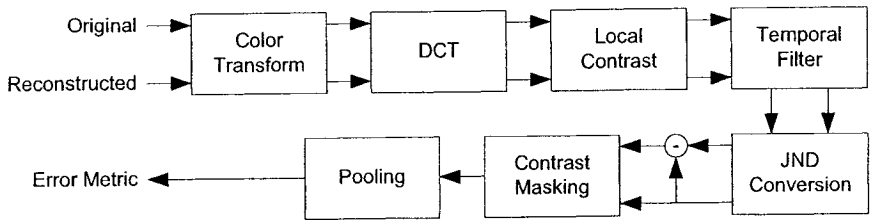



Figure 2.10: Block diagram of Watson's DVQ [4]

The purpose of each component in Figure 2.10 is listed as follows.

- *Color transformation*: Because the human retina transforms colors into an opponent color system [69], DVQ uses YOZ [4] color space instead of YCbCr color space.
- *DCT*: Because the HVS treats different frequencies differently, DVQ uses DCT to transform input to the frequency domain.
- *Local Contrast*: The HVS can only detect patterns if the amplitude difference is large for bright region. A low-pass time-filter is used to model this effect.
- *Temporal Filter*: The temporal filter is a low-pass filter in the temporal direction because HVS reacts like a low-pass filter for high temporal frequencies.
- *Just-Noticeable-Difference (JND) Conversion*: Each frequency component is divided by the corresponding spatial threshold defined in [70] because the HVS reacts differently for different frequencies.
- *Contrast Masking*: The visibility of a pattern is reduced by the presence of other components in the image, especially the ones that appear at the same location and share the same spatial frequency.
- *Pooling*: Pooling combines all JNDs to obtain the error metric (one number).

Note that Watson's DVQ is significantly more computationally intensive than PSNR- or MSE-based statistical quality measures. However, it should give a better match to the perceptual quality because it takes the characteristics of the HVS into account.

## 2.3. Digital Video Transcoding

In many communication applications, one would like to change the encoded video format or to change the bitrate of a compressed bitstream. In such situations, transcoding can be used.

Video transcoding is the process of converting a compressed video format into another compressed video format. Transcoding is required for transmitting pre-encoded video on a channel that has lower bandwidth or is congested. It is used in many emerging services such as video-on-demand (VOD), digital terrestrial television broadcasting (DTTB), and cable television (CATV) distribution [71]. In such applications, videos stored in the database are compressed at high quality. Depending on the consumer channel bandwidth and/or the quality requirement, transcoder is used to bring the high-quality video to the desired lower-quality video.

Generally, transcoding is used in the following scenarios: a) when the bitrate of a compressed bitstream needs to be reduced, or b) when the video compression format needs to be changed from one standard to another, or c) a combination of both a) and b). For bitrate reduction, there are many methods to achieve the goal: requantization, spatial resolution reduction, temporal resolution reduction, or any combination of these methods. The transcoding schemes studied in this thesis deal with bitrate reduction by requantizing the compressed bitstreams.

Two important aspects related to video transcoding are discussed here. Section 2.3.1 discusses the advantage and disadvantage of several transcoder architectures. In Section 2.3.2, the requantization error introduced by transcoding is also discussed.

### 2.3.1. Transcoder Architectures

Although there are many transcoder architectures, rather than fully decompress and recompress the video sequence, which requires computationally intensive motion estimation, the original motion vectors are reused in all of them.

The easiest transcoding can be achieved by directly re-quantizing the DCT coefficients with coarser quantizer. This transcoder architecture is often referred as open-loop architecture [72]. As shown in Figure 2.11, input bitstream is first VLC decoded (VLD) to obtain the quantized DCT coefficients. These quantized coefficients are first dequantized ($Q_1^{-1}$) with step size $Q_1$, then requantized with a new quantization step size $Q_2$. The requantized DCT coefficients are then variable-length coded to produce the output bitstream.

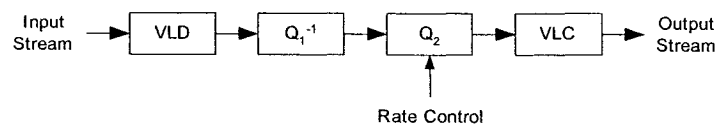



Figure 2.11: Open-loop transcoder architecture using requantization [72]

One problem with the open-loop architecture is that it does not take drift error into account. Drift error happens because, after requantizing the DCT coefficients, the residues are no longer accurate due to possible changes (due to requantization) in the

reference frame(s). Drift correction can be done either in the spatial domain [73][74] or in the frequency domain [75]. As pointed out in [72], frequency domain drift correction usually requires a higher computational complexity. Thus, only pixel-domain transcoder is studied in more detail.

Several pixel-domain transcoder architectures with drift correction have been proposed in the past. The most straightforward transcoder is the cascaded decoder-encoder architecture [72] [76]. As shown in Figure 2.12, the decoded motion vectors are used to encode the picture.

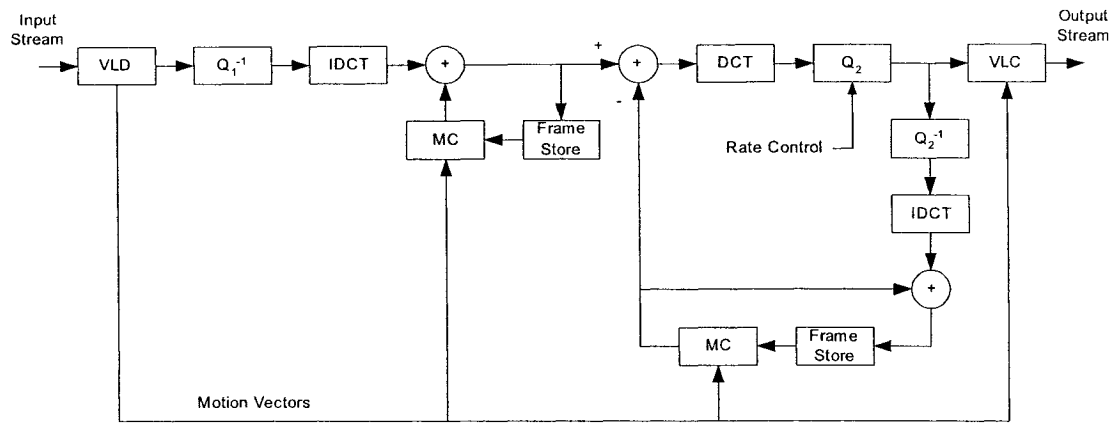



Figure 2.12: Cascaded decoder-encoder pixel-domain transcoder architecture [72][76]

To further reduce the computational complexity, two fast pixel-domain transcoder architectures were proposed in [73] and [75]. Figure 2.13 shows an example of a fast drift-correction pixel-domain transcoder architecture. Note, although these architectures can compensate most drift errors caused by requantization, as analyzed in [72][77][78], they also introduce a small amount of drift errors. These drift errors are caused by several reasons, such as clipping before the frame memory, arithmetic precision DCT/IDCT mismatch, and linear assumption of the MC [78].

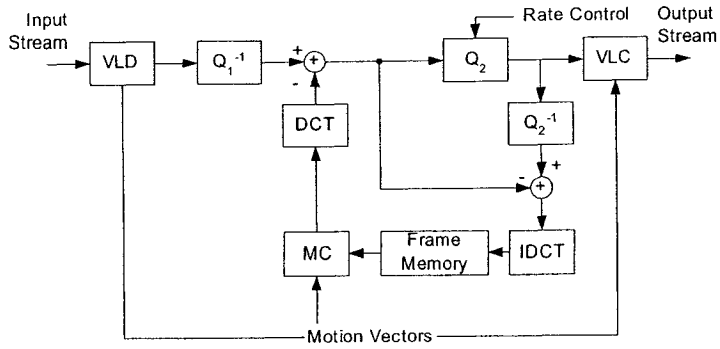



Figure 2.13: Fast pixel-domain transcoder architecture with drift error correction [73]

Since the main focus of this thesis is to achieve constant quality, it is essential that a CQ transcoder eliminates the drift error completely. As a result, the cascaded decoder-encoder transcoder architecture shown in Figure 2.12 is used in this thesis.

### 2.3.2. Requantization Errors Introduced by Transcoding

In Figure 2.12, it is assumed that the input bitstream was quantized with quantization step size $Q_1$. The transcoder needs first to dequantize the bitstream with the same step size ($Q_1$). After the drift correction, the MB is then requantized with step size $Q_2$. In general, requantizing the input with a different step size ($Q_2$) introduces error.

As shown in Figure 2.14, requantization may introduce two types of error: the typical quantization error, and the requantization error. Cross signs are used for break points, and circles are used for reconstruction points. The subscript indicates the quantizer used.

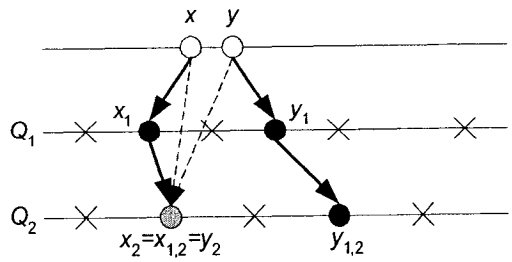



Figure 2.14: Normal quantization error and Requantization error

Typical quantization error can be observed for original signal $x$. As shown in Figure 2.14, $x$ is first quantized with $Q_1$, and is dequantized to $x_1$. $x_1$ is then quantized with $Q_2$, and has its dequantized value to be $x_{1,2}$. Note that even after two quantizations, the final dequantized value ($x_{1,2}$) is the same as if $x$ is directly quantized with $Q_2$.

However, for value $y$, the final dequantized value ($y_{1,2}$) is different from direct quantization ($y_2$) with $Q_2$. This inconsistency is called requantization error. As shown in [79], the requantization error can be reduced by estimating the Laplacian parameters of DCT coefficients. More information about transcoding can be found in [74]-[81].

## 2.4. Chapter Summary

In this chapter, different technologies used in digital video compression standards (MPEG-1/2/4 mainly) are reviewed. The main principle of video compression is to reduce the temporal redundancy with ME/MC, and reduce the spatial redundancies with DCT followed by heavy quantization. The quantized blocks are then entropy coded by using run-length coding and variable-length coding to remove coding redundancies.

Common distortions and artifacts typically present in the compressed videos are also examined. These distortions are measured quantitatively by using video quality

metrics, which can be either subjective or objective. Besides using PSNR as the objective video quality measure, subjective quality predictors could also be used. Watson's DVQ is a subjective quality predictor that models several characteristics of the human visual system.

Some advanced topics and applications of digital video compression are also reviewed, such as error resilience and parallel encoding/decoding. These technologies are used to enhance the robustness, security, and performance of video compression systems. The purposes and principles behind transcoding are also reviewed. These technologies allow dynamically changing the compressed video format. Many emerging communication services, such as video on demand and digital cable television, are based on them.

Digital video compression is an ongoing research area. With advances made by researchers, higher compression will be achieved. This in turn leads to better video quality, and, most importantly, more efficient use of channel bandwidth and storage space. All of these translate to more profit to content/service providers, and better/cheaper services for consumers.

# Chapter 3

# MPEG-4 Laplacian Rate Models and Distortion Models

In making rate control decisions, it would be very useful to have a rate model and a distortion model for the video data being compressed. This chapter develops distortion models that relate the QP (quantization parameter) chosen to the expected error that will result from that choice. They are parameterized by QP. Similarly, the rate models relate the QP chosen to the expected number of bits. Because the encoding difficulties change for each frame and for each sequence, the models also have to adapt to input characteristics changes. Thus, the rate/distortion (R/D) models developed in this chapter are data dependent. That is, the same QP decision would result in different expected error and expected number of bits for different frames and sequences.

Several approaches for rate/distortion modeling have been proposed in the past. They can be classified as empirical ([82],[83],[84]) and analytical approaches. The empirical approach predicts the rate/distortion by mathematical processing of the observed R/D data. Interpolation is usually used to predict missing data points. As an example, Tsai [82] uses a dynamic rate table to predict the number of bits. The table contains the number of bits of the previously coded macroblocks (MBs), and is indexed by the complexity measured in SAD (sum of absolute difference) and QP. Although the empirical approach tends to use fewer computations than the analytical approach, in general, its performance degrades for non-temporally-stationary sequences.

The analytical approach assumes that AC (alternate current) DCT (discrete cosine transform) coefficients have certain statistical distribution. Four ways to model and analyze the AC DCT coefficients have been proposed in the past: Gaussian distribution ([85],[86]), generalized Gaussian and Laplacian distributions ([7],[87]-[94]), the $\rho$-domain analysis ([95][96]), and centralized Cauchy density ([97]). Among these, the Laplacian distribution is the most widely used model in practice. Hence, it is also used in the rate/distortion models developed in this chapter.

Although so many rate/distortion models are reported, and many of them work well, they share two common issues. First, the model parameters are usually computed from historical records. This leads to a problem similar to that encountered in the empirical approach: the performance of the model degrades if sequences are not temporally stationary. Another common problem is the assumption that the quantizer is just a uniform quantizer. This assumption is true to some extent, but is not entirely true. In general, mid-riser uniform quantizers are used for quantizing intra frames, and mid-step uniform quantizers are used to quantize intra frames. However, in video compression, dead-zone is applied to all quantizers. Having dead-zone would increase the probability of having DCT coefficients quantized to zero. It also alters the position of the break-points and reconstruction points of the quantizers. Hence, ignoring the dead-zone effect would degrade the accuracy of these models.

The rate/distortion models developed in this chapter assume the DCT coefficients have Laplacian distributions. To address the first issue mentioned in the previous paragraph, recall that the Laplacian distributions are controlled by the Laplacian parameters. These parameters are derived from the current input frame in the proposed

rate/distortion model. Thus, the proposed models neither assume nor require temporal stationarity of the input sequence. To address the second issue, a generalized uniform quantizer with dead-zone is used in the derivations. Thus, the effect of the dead-zone is considered for both intra and inter frames.

Two sets of rate/distortion models for the MPEG-4 standard are developed in this chapter: one set for the MPEG-4 encoder, and another set for the MPEG-4 transcoder. The rate/distortion models for MPEG-4 encoding are first presented in Section 3.1. They are parameterized by the Laplacian parameter estimated from the current input DCT coefficients. As for the transcoder, since its input is already quantized, the input coefficients no longer have the Laplacian distribution. Thus, a similar but different rate/distortion models for MPEG-4 transcoding are developed in Section 3.2. Section 3.3 summarizes the chapter.

## 3.1. Laplacian Rate Model and Distortion Model for MPEG-4 Video Encoding

In this section, the Laplacian distortion model for MPEG-4 encoding is first developed in Section 3.1.1. This model is developed by simply passing the Laplacian distributed DCT coefficients through a generalized uniform quantizer. Since the number of bits to compress a frame is directly related to how the quantized DCT coefficients are entropy coded, information about the MPEG-4 entropy coder is needed. A model for this entropy coder is proposed in Section 3.1.2. It is then used later to develop the Laplacian rate model in Section 3.1.3. Since both the rate and distortion models are parameterized

by the Laplacian parameter, the method to estimate the Laplacian parameter is presented in Section 3.1.4. Finally, some simulation results are presented in 3.1.5 to verify the correctness of the proposed models.

### 3.1.1. Laplacian Distortion Model for MPEG Video Encoding

In general, DCT coefficients can be modeled as having Laplacian or Gaussian distributions. Both models are widely used. The rate-distortion models presented in this chapter assume DCT coefficients have Laplacian distributions [9][98] controlled by the Laplacian parameters ($\alpha$):

$$f(x)=\frac{\alpha}{2}e^{-\alpha|x|} \tag{8}$$

For intra-frames, the distortion comes from the quantization noise when DCT coefficients are quantized. For inter-frames, the distortion comes from the quantization noise of the DCT coefficients of the residues. In both cases, the quantizers have dead-zones. Because the intra-quantizer has a different dead-zone size than the inter-quantizer, a generalized uniform quantizer with dead-zone is used in the analysis as shown in Figure 3.1, where $Q$ is the quantization step-size, $\gamma$ is the location of the first break point, and $\rho$ is the location of the first reconstruction level. In MPEG-4, when the H.263 quantizer is used, $\gamma$=1 and $\rho$=3/2 are used for intra quantizer, and $\gamma$=5/4 and $\rho$=3/2 are used for inter quantizer.

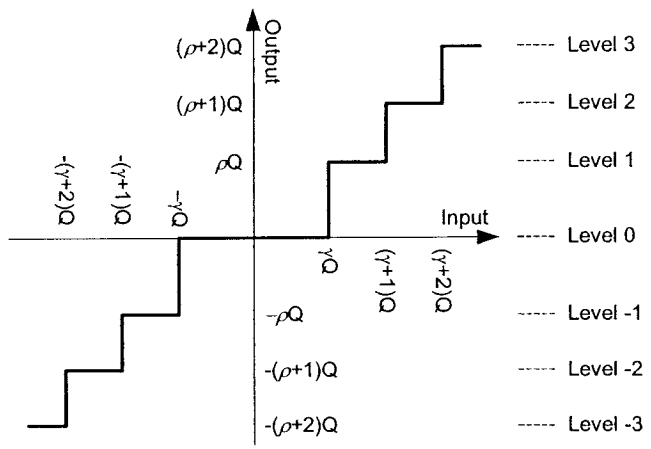


Figure 3.1: Generalized uniform quantizer with dead-zone [99]

By passing the DCT coefficients that are modeled as Laplacian random variables through the quantizer shown in Figure 3.1, the probability density function (PDF) $f_N$ of the quantization noise ($x$) can be obtained as [99]:

- For $\gamma \le \rho \le \gamma + \frac{1}{2}$:
  - If $|x| \le (\rho - \gamma)Q$:

$$f_N(x \mid \alpha) = \overbrace{\frac{\alpha}{2} e^{-\alpha|x|}}^{k=0} + \overbrace{\sum_{k=1}^{\infty} \left[ \frac{\alpha}{2} e^{-\alpha[(\rho+k-1)Q+x]} + \frac{\alpha}{2} e^{\alpha[-(\rho+k-1)Q+x]} \right]}^{k\neq 0} \tag{9}$$
$$= \frac{\alpha}{2} e^{-\alpha|x|} + \frac{\alpha}{2} \frac{e^{-\alpha\rho Q}}{1 - e^{-\alpha Q}} \left( e^{-\alpha x} + e^{\alpha x} \right)$$

  - If $(\rho - \gamma)Q < |x| \le (\gamma - \rho + 1)Q$:

$$f_N(x \mid \alpha) = \frac{\alpha}{2} e^{-\alpha|x|} + \sum_{k=1}^{\infty} \frac{\alpha}{2} e^{-\alpha[(\rho+k-1)Q+|x|]} = \frac{\alpha}{2} e^{-\alpha|x|} + \frac{\alpha}{2} \frac{e^{-\alpha\rho Q}}{1 - e^{-\alpha Q}} e^{-\alpha|x|} \tag{10}$$

  - If $(\gamma - \rho + 1)Q < |x| \le \gamma Q$:

$$f_N(x \mid \alpha) = \frac{\alpha}{2} e^{-\alpha|x|} \tag{11}$$

Equation (12) summarizes equations (9), (10), and (11):

$$f_N(x\,|\,\alpha)=\begin{cases}\dfrac{\alpha}{2}e^{-\alpha|x|}+\dfrac{\alpha}{2}\dfrac{e^{-\alpha\rho Q}}{1-e^{-\alpha Q}}\left(e^{-\alpha x}+e^{\alpha x}\right) & |x|\le(\rho-\gamma)Q\\ \dfrac{\alpha}{2}e^{-\alpha|x|}+\dfrac{\alpha}{2}\dfrac{e^{-\alpha\rho Q}}{1-e^{-\alpha Q}}e^{-\alpha|x|} & (\rho-\gamma)Q<|x|\le(\gamma-\rho+1)Q\\ \dfrac{\alpha}{2}e^{-\alpha|x|} & (\gamma-\rho+1)Q<|x|\le\gamma Q\\ 0 & |x|>\gamma Q\end{cases} \tag{12}$$

- For $\gamma+\frac{1}{2}<\rho\le\gamma+1$:
  - If $|x|\le(\gamma-\rho+1)Q$:

$$\begin{aligned}f_N(x\,|\,\alpha)&=\overbrace{\frac{\alpha}{2}e^{-\alpha|x|}}^{k=0}+\overbrace{\sum_{k=1}^{\infty}\left[\frac{\alpha}{2}e^{-\alpha\left[(\rho+k-1)Q+x\right]}+\frac{\alpha}{2}e^{\alpha\left[-(\rho+k-1)Q+x\right]}\right]}^{k\neq 0}\\&=\frac{\alpha}{2}e^{-\alpha|x|}+\frac{\alpha}{2}\frac{e^{-\alpha\rho Q}}{1-e^{-\alpha Q}}\left(e^{-\alpha x}+e^{\alpha x}\right)\end{aligned} \tag{13}$$

  - If $(\gamma-\rho+1)Q<|x|\le(\rho-\gamma)Q$:

$$f_N(x\,|\,\alpha)=\frac{\alpha}{2}e^{-\alpha|x|}+\sum_{k=1}^{\infty}\frac{\alpha}{2}e^{-\alpha\left[(\rho+k-1)Q-|x|\right]}=\frac{\alpha}{2}e^{-\alpha|x|}+\frac{\alpha}{2}\frac{e^{-\alpha\rho Q}}{1-e^{-\alpha Q}}e^{\alpha|x|} \tag{14}$$

  - If $(\rho-\gamma)Q<|x|\le\gamma Q$:

$$f_N(x\,|\,\alpha)=\frac{\alpha}{2}e^{-\alpha|x|} \tag{15}$$

Equation (16) summarizes equations (13), (14), and (15):

$$f_N(x\,|\,\alpha)=\begin{cases}\dfrac{\alpha}{2}e^{-\alpha|x|}+\dfrac{\alpha}{2}\dfrac{e^{-\alpha\rho Q}}{1-e^{-\alpha Q}}\left(e^{-\alpha x}+e^{\alpha x}\right) & |x|\le(\gamma-\rho+1)Q\\ \dfrac{\alpha}{2}e^{-\alpha|x|}+\dfrac{\alpha}{2}\dfrac{e^{-\alpha\rho Q}}{1-e^{-\alpha Q}}e^{\alpha|x|} & (\gamma-\rho+1)Q<|x|\le(\rho-\gamma)Q\\ \dfrac{\alpha}{2}e^{-\alpha|x|} & (\rho-\gamma)Q<|x|\le\gamma Q\\ 0 & |x|>\gamma Q\end{cases} \tag{16}$$

As an example, for the H.263 intra-quantizer ($\gamma$=1, $\rho$=3/2), and $Q$=10, the PDF of the quantization noise for different Laplacian parameters ($\alpha$) are shown in Figure 3.2.

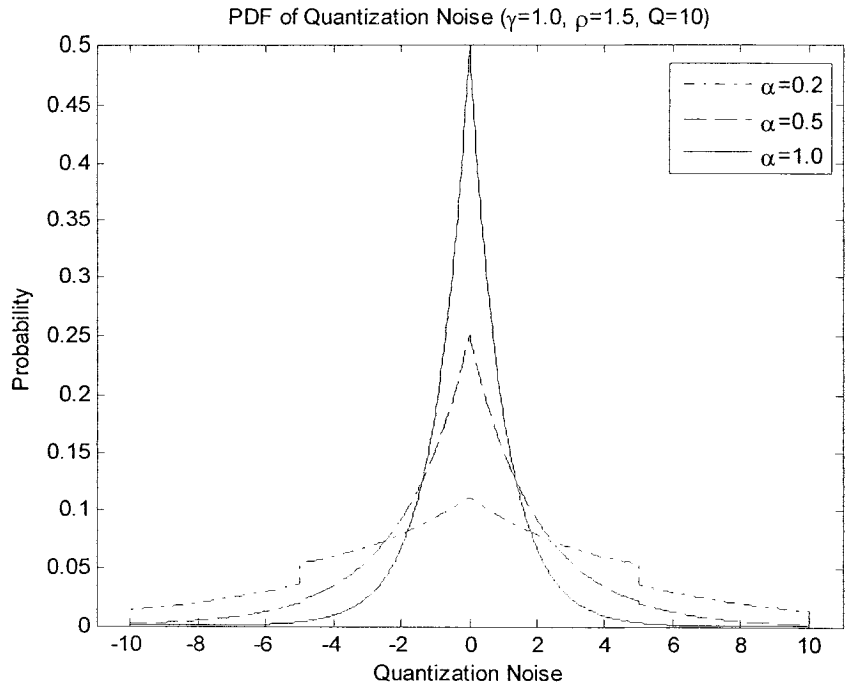



Figure 3.2: PDF of the Quantization Noise of a DCT coefficient for $\gamma$=1, $\rho$=3/2, and Q=10

In both cases, the quantization noise is measured in mean-squared error (MSE). The quantization noise as a function of quantization steps-size ($Q$) can be shown to be

$$\begin{aligned} & MSE(Q \mid \alpha) \\ & = E_N\left[X^2\right] = \int_{-\infty}^{\infty} x^2 f_N(x \mid \alpha) dx \\ & = \frac{2}{\alpha^2} - \left(\gamma^2 Q^2 + \frac{2\gamma}{\alpha} Q + \frac{2}{\alpha^2}\right) e^{-\alpha\gamma Q} + \\ & \frac{e^{-\alpha\gamma Q}}{1-e^{-\alpha Q}}\left[(\gamma-\rho)^2 Q^2 + \frac{2(\gamma-\rho)}{\alpha} Q + \frac{2}{\alpha^2}\right] - \\ & \frac{e^{-\alpha(\gamma+1)Q}}{1-e^{-\alpha Q}}\left[(\gamma-\rho+1)^2 Q^2 + \frac{2(\gamma-\rho+1)}{\alpha} Q + \frac{2}{\alpha^2}\right] \end{aligned} \tag{17}$$

As an example, for intra-quantizer ($\gamma$= 1, $\rho$= 3/2), the MSEs and the PSNRs (peak signal-to-noise ratio) for different Laplacian parameters ($\alpha$) are shown in Figure 3.3 and Figure 3.4. Note that as the quantization step-size increases, PSNR decreases until saturation occurs (all quantized coefficients become zero). Similarly, as the Laplacian parameter increases, DCT coefficients become more concentrated around zero (smaller). Hence, the quantization noise (even if saturation occurs) becomes smaller (larger PSNR) for the same QP.

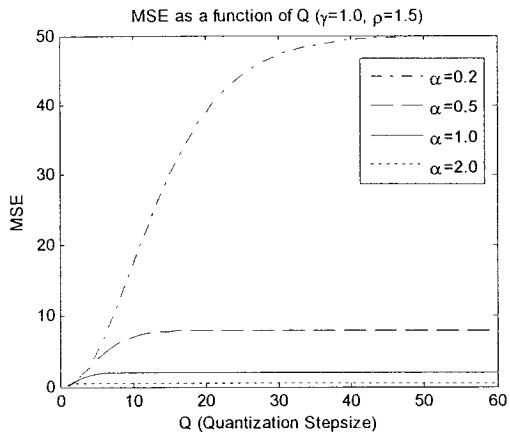


Figure 3.3: MSE as a function of QP

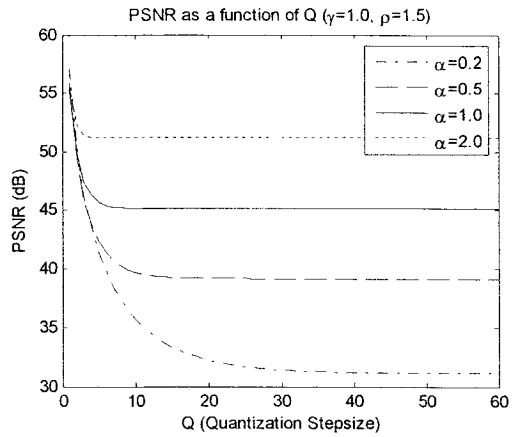


Figure 3.4: PSNR as a function of QP

## 3.1.2. Modeling the MPEG-4 Variable Length Coding (VLC) Table

In MPEG-4, quantized AC DCT coefficients are scanned into one dimension. Joint symbols are then formed by combining one non-zero AC coefficient with a run of zero-value AC coefficients. The non-zero AC coefficient is specified by its sign and its level. The *level* is proportional to the reconstruction magnitude (see Figure 3.1). Instead of estimating the bits for compound DCT coefficients, to simplify the analysis, the cost of each AC coefficient is estimated alone.

To do this simplification, from the MPEG-4 document, the lengths of VLC codewords that have no preceding zero coefficients (the most likely case) are first examined. As shown in Figure 3.5, the actual VLC codeword length for a single non-zero AC coefficient can be approximated with a linear model.

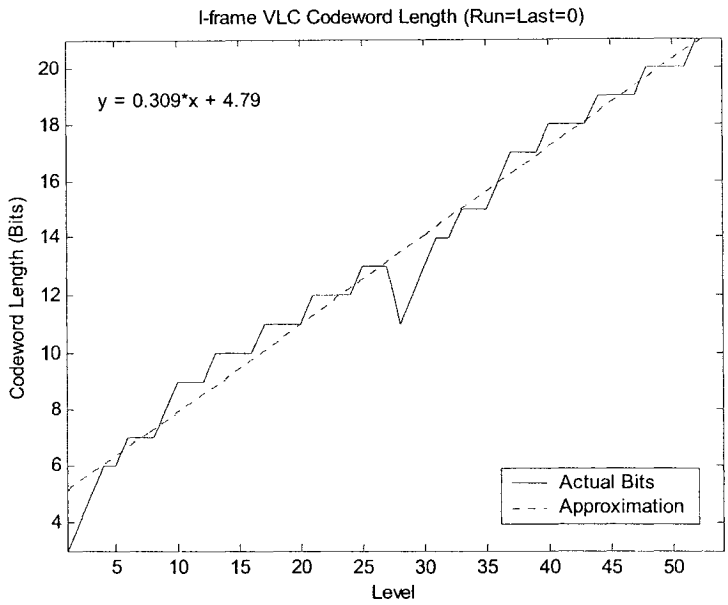



Figure 3.5: Codeword length of MPEG-4 VLC table as a function of levels [99]

For AC coefficients quantized to zero, Figure 3.6 shows the codeword length for *level*=1 (the most likely non-zero level) versus the number of consecutive zeros (run). Again, a linear relationship is found. Similar behaviours can be observed for *level*>1. Thus, this implies that each zero AC coefficient can be considered as having a fixed number of bits.

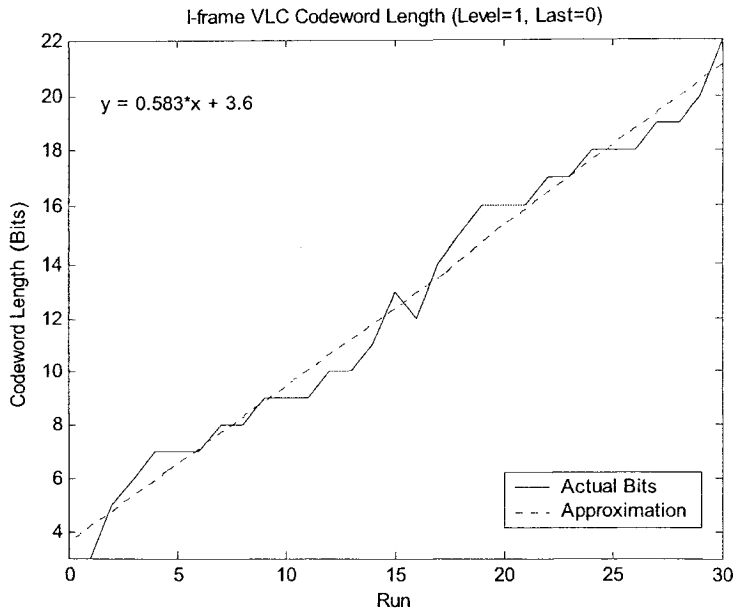



Figure 3.6: Codeword length of MPEG-4 VLC table as a function of runs [99]

Because of the limited VLC table size, the MPEG-4 VLC table cannot index all possible combinations of levels and runs. For levels and runs that could not be covered by the table, a fixed-length code (type 3 escape code) is used.

To summarize, the bits ($r$) per level ($k$) of the MPEG-4 VLC table can be modeled as [99]:

$$r(k)=\begin{cases} z & k=0 \\ r_{VLC}(k)=a|k|+b & k\neq 0,|k|\leq M \\ r_{FLC}(k)=c & k\neq 0,|k|>M \end{cases} \tag{18}$$

The first equation represents bits for zero AC coefficients, the second equation represents bits for non-zero AC coefficients defined in the VLC table, and the third equation represents the bits for fixed-length code. In the simulation, the values for $a$, $b$, $c$, $z$ and $M$ are listed in Table 3.1.

Table 3.1: Model Parameters for MPEG-4 VLC Table [99]

| | $a$ | $B$ | $c$ | $z$ | $M$ |
|---|---|---|---|---|---|
| Intra Block | 0.3092 | 4.7932 | 30 | 0.05831 | 54 |
| Inter Block | 0.6904 | 4.5362 | 30 | 0.03294 | 24 |

### 3.1.3. Laplacian Rate Model for MPEG-4 Video Encoding

By passing the two-sided Laplacian distributed input through the generalized quantizer (see Figure 3.1), the output level ($k$) has the probability density function $f_L(k)$ as

- For $k=0$: $f_L(k=0\mid\alpha)=2\int_0^{\gamma Q}\frac{\alpha}{2}e^{-\alpha x}dx=1-e^{-\alpha\gamma Q}$
- For $k\neq 0$: $f_L(k\neq 0\mid\alpha)=\int_{(\gamma+|k|-1)Q}^{(\gamma+|k|)Q}\frac{\alpha}{2}e^{-\alpha x}dx=\frac{e^{-\alpha\gamma Q}\left(e^{\alpha Q}-1\right)}{2}e^{-\alpha Q|k|}$

Thus, the quantized DCT level ($k$) has PDF as

$$f_L(k\mid\alpha)=\begin{cases}1-e^{-\alpha\gamma Q} & k=0\\ \dfrac{e^{-\alpha\gamma Q}\left(e^{\alpha Q}-1\right)}{2}e^{-\alpha Q|k|} & k\neq 0\end{cases} \tag{19}$$

With the knowledge of the PDF of each quantized DCT level and the bit cost of each level, the expected number of bits to code a DCT coefficient (with Laplacian parameter $\alpha$) when quantized by $Q$ is

$$\begin{aligned}Bit(Q\mid\alpha)&=E[Bit]=\sum_{k=-\infty}^{\infty}r(k)f_L(k\mid\alpha)\\ &=z\left(1-e^{-\alpha\gamma Q}\right)+2\sum_{k=1}^{M}(ak+b)\left(\frac{e^{-\alpha\gamma Q}\left(e^{\alpha Q}-1\right)}{2}e^{-\alpha Qk}\right)+2\sum_{k=M+1}^{\infty}c\left(\frac{e^{-\alpha\gamma Q}\left(e^{\alpha Q}-1\right)}{2}e^{-\alpha Qk}\right)\\ &=z+(a+b-z)e^{-\alpha\gamma Q}-(aM+b-c)e^{-\alpha Q(M+\gamma)}+a\frac{e^{-\alpha Q(\gamma+1)}-e^{-\alpha Q(M+\gamma)}}{1-e^{-\alpha Q}}\end{aligned} \tag{20}$$

As an example, Figure 3.7 shows the bits per DCT coefficient as a function of quantization parameter $Q$ for different Laplacian parameters ($\alpha$) when quantized with the H.263 intra quantizer. As expected, when $\alpha$ increases, the DCT coefficients are more concentrated around zero, hence the number of bits decreases. When $Q$ increases, the number of bits decreases.

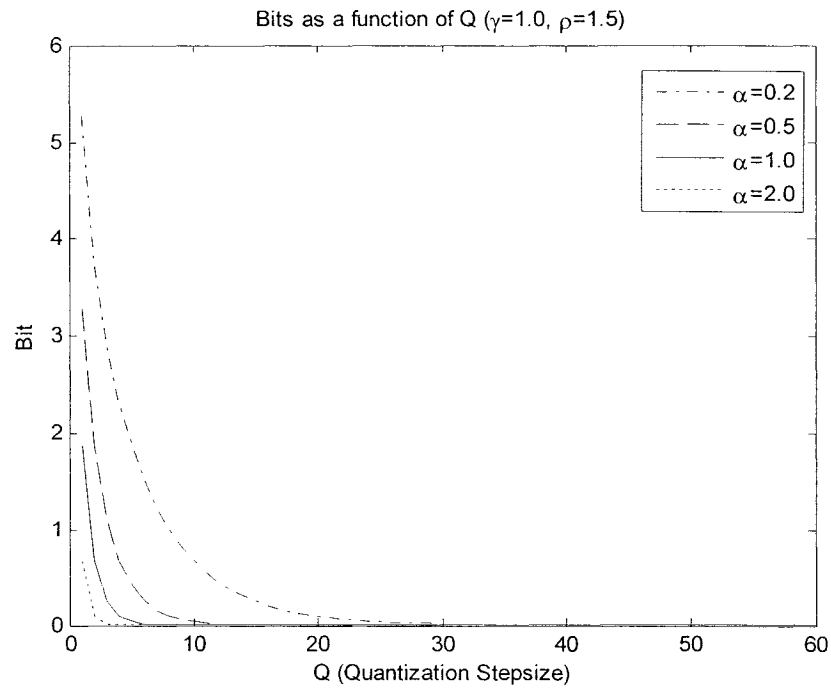



Figure 3.7: Bits per DCT coefficient as function of $Q$ for H.263 intra quantizer

### 3.1.4. Laplacian Parameter Estimation for MPEG Video Encoding

Assuming AC coefficients in different DCT blocks are independent, and that they have Laplacian distributions with the same parameter for each AC frequency, then there are 63 Laplacian parameters ($\alpha_{i,j}$) to be estimated for intra frames, and 64 for inter frames. Using the maximum-likelihood estimation theory [100], $\alpha$ for DCT frequency $(i,j)$ can be obtained as

$$\alpha_{i,j} = \frac{1}{E\left[\left|dct_{i,j}\right|\right]} \cong \frac{M}{\sum_{m=1}^{M}\left|dct_{i,j}[m]\right|} \tag{21}$$

where $dct_{i,j}[m]$ is the DCT coefficient of frequency $(i,j)$ in DCT block $m$, $M$ is the number of blocks (size of sample space) used in the estimation, and $i,j = 0\ldots7$. In this thesis, *dct* is chosen to be the set of all DCT coefficients over the entire current frame (or the difference frame) to be coded, i.e., $M$ is the number of DCT blocks in the frame.

Traditionally, rate-control algorithms ([101]-[107]) predict the rate/distortion of the current frame using past information. For example, [101] keeps historical data containing the rate and distortion of the previously encoded frames. QPs are calculated from this historical record. This implicitly assumes temporal stationarity among frames. When the video is not temporally stationary, the model may generate bad rate/distortion prediction, which may lead the rate-control algorithm to under/over-spend the bits and decrease/increase the quality.

Unlike traditional rate-control algorithms that estimate rate/distortion based on the historical record, the Laplacian distortion model does not rely on past information. Only the DCT coefficients of the current frame are needed. Thus, the Laplacian model will give a consistent performance for all frames without requiring any side information about stationarity of the frame. This feature allows better response to changes in the sequence characteristics such as scene change, high motion scenes, zoom in/out, etc.

Note that for stationary scenes, it is possible to improve the prediction accuracy further by estimating the Laplacian parameters using DCT coefficients from multiple frames. However, this would increase the computational complexity as well. As demonstrated in the next section, the Laplacian rate/distortion model can give good prediction based on the Laplacian parameters estimated from the current frame only.

### 3.1.5. Verification of the Laplacian Rate-Distortion Model

To verify whether the proposed model predicts rate and distortion accurately, simulations are performed using an MPEG-4 encoder with constant QP for three 150-frame test sequences (table tennis, football, and flower garden). All frames are intra coded.

The analytical (using the Laplacian rate-distortion model) and empirical (using constant QP) average rate/distortion results for a single DCT block are shown in Table 3.2. For sequence *football*, the rate and distortion curves of each frame for various QPs are also shown in Figure 3.8 and Figure 3.9 respectively. These results indicate that the analytical rates and distortions predicted by the proposed model follow the simulation results very closely. Note that the $y$ axes of these figures are at different scales to better highlight the differences between the analytical and empirical curves for each QP.

Table 3.2: Comparison between Simulation (S) and Analytical (A) Rate/Distortion results [108]

| | Table Tennis | | | | Football | | | | Flower Garden | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rate (bit) | | PSNR (dB) | | Rate (bit) | | PSNR (dB) | | Rate (bit) | | PSNR (dB) | |
| QP | S | A | S | A | S | A | S | A | S | A | S | A |
| 1 | 309.54 | 322.73 | 49.92 | 50.85 | 310.21 | 310.83 | 49.97 | 50.61 | 456.75 | 474.52 | 50.20 | 51.54 |
| 7 | 67.76 | 78.85 | 34.21 | 33.01 | 70.70 | 74.20 | 34.40 | 33.73 | 148.36 | 152.76 | 33.84 | 32.93 |
| 13 | 33.31 | 36.04 | 30.65 | 28.91 | 35.21 | 36.61 | 30.50 | 29.67 | 91.92 | 88.99 | 28.55 | 27.74 |
| 19 | 20.70 | 21.02 | 28.77 | 27.00 | 21.34 | 21.98 | 28.44 | 27.58 | 63.90 | 58.14 | 25.61 | 24.91 |
| 25 | 14.59 | 14.51 | 27.49 | 25.91 | 14.45 | 14.75 | 27.07 | 26.27 | 46.75 | 40.41 | 23.67 | 23.08 |
| 31 | 11.08 | 11.40 | 26.53 | 25.22 | 10.48 | 10.67 | 26.04 | 25.37 | 35.21 | 29.33 | 22.26 | 21.79 |

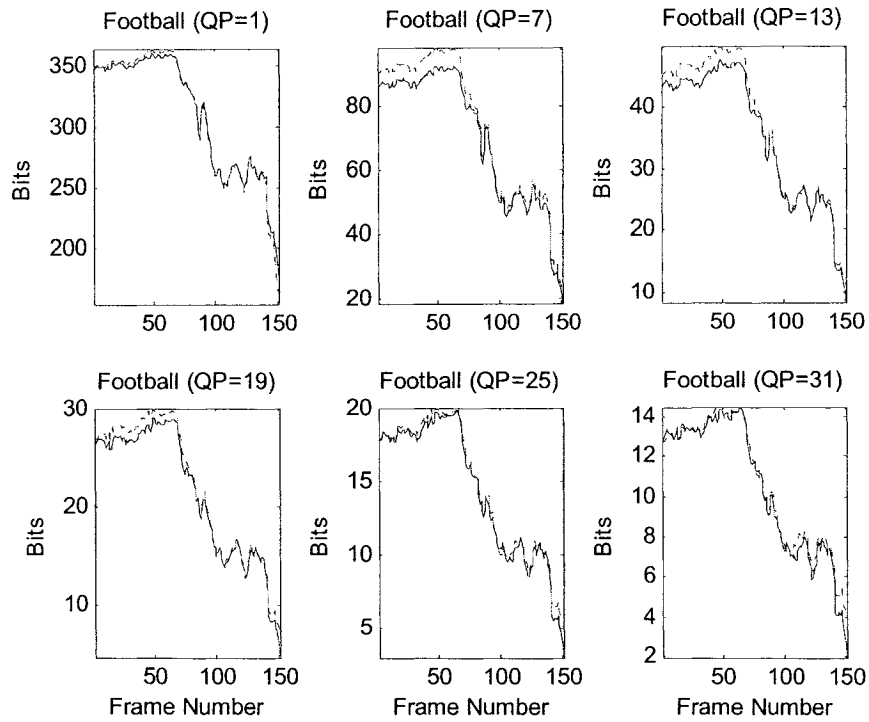



Figure 3.8: Rate of Football for various QPs
(Solid curve: Simulation, Dashed curve: Analytical, estimated using the Laplacian rate model) [108]

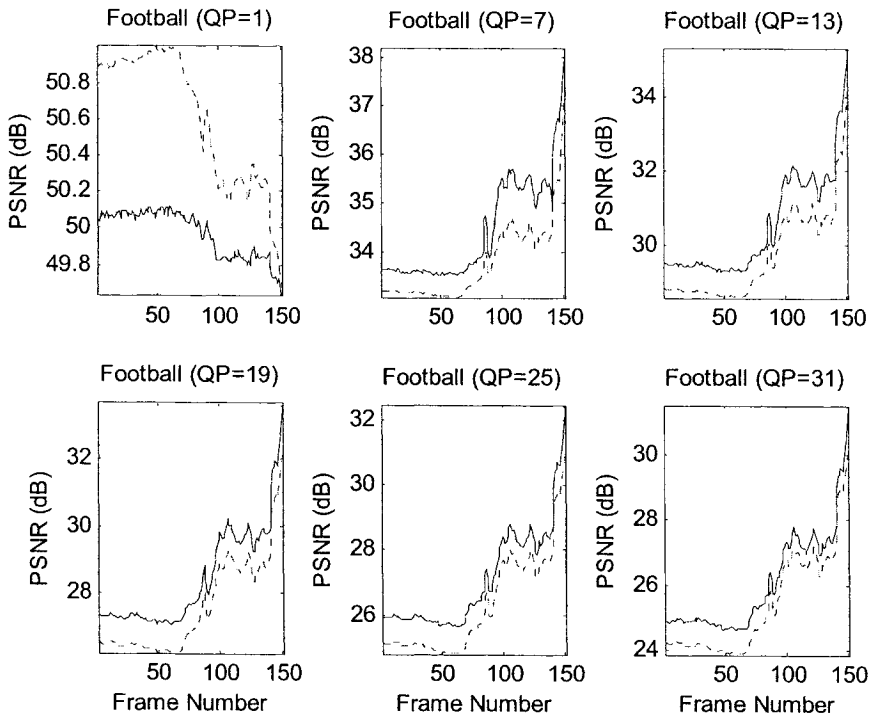



Figure 3.9: PSNR of Football for various QPs
(Solid curve: Simulation, Dashed curve: Analytical, estimated using the Laplacian distortion model) [108]

## 3.2. Laplacian Rate Model and Distortion Model for MPEG-4 Video Transcoding

In the previous section, the Laplacian Rate/Distortion models are proposed for MPEG-4 video encoder. For an encoder, since the original video sequence is accessible, the exact values of the unquantized DCT coefficients are known. However, for a transcoder, the situation becomes different. The input of a transcoder is an encoded bitstream (see Figure 3.10), which contains quantized DCT coefficients. Since quantization introduces noise, transcoders no longer have a good knowledge about the original sequence. In some situations, heavy quantization of the incoming bitstream causes many DCT coefficients to be quantized to zero. Thus, all information about the original sequence is lost. Due to the noise present in the transcoder input bitstream, the transcoding R/D model is expected to be less accurate than the encoding R/D models.

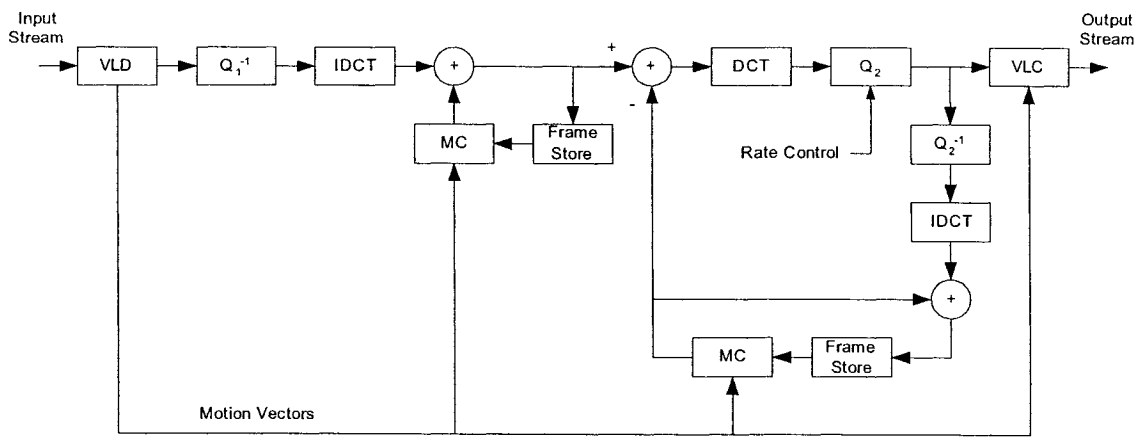



Figure 3.10: Cascaded decoder-encoder pixel-domain transcoder architecture [72][76]

In transcoder, the input contains quantized DCT coefficients rather than unquanitzed DCT coefficients. This means that the Laplacian R/D model developed in

the previous section for encoding cannot be applied directly for transcoding. However, the same principle still applies to transcoding. As a result, although the original unquanitzed input is not accessible at the transcoder, it is still valid to use the same assumption as the encoding case that the unquantized DCT coefficients have Laplacian distributions.

Another important aspect in transcoding is that the input will be re-quantized. As discussed in Section 2.3.2, requantization introduces additional error than direct quantization. Thus, this effect should be handled by the rate-distortion model for transcoding. As shown in Figure 3.10, the input was quantized with step size $Q_1$, and is first dequantized with $Q_1$. The drift corrected signal is then quantized with step size $Q_2$. Depending on the relative size of $Q_1$ and $Q_2$, different relationships of the break/reconstruction points between the first and the second quantizers are shown in Figure 3.11 and Figure 3.12. The break points are shown as crosses, and the reconstruction points are shown as circles. $B_{m,n}/R_{m,n}$ are used to denote the $n$-th break/reconstruction point of the quantizer with step size of $Q_m$.

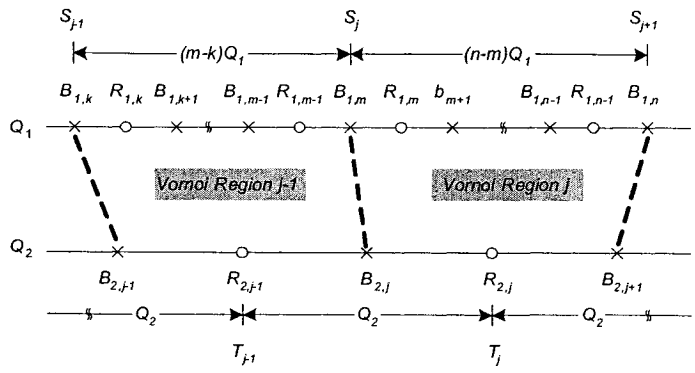



Figure 3.11: Break and reconstruction points of $Q_1$ and $Q_2$ in transcoder ($Q_1<Q_2$)

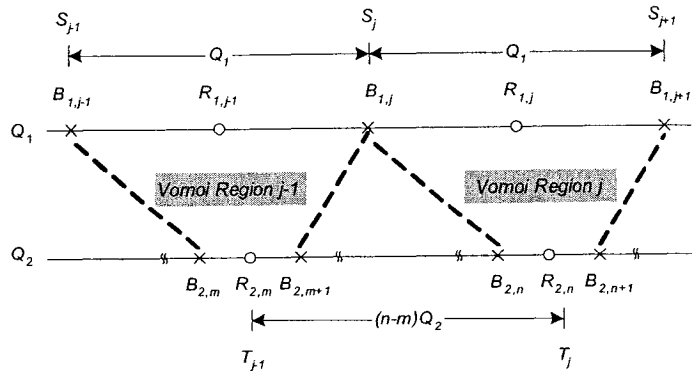



Figure 3.12: Break and reconstruction points of $Q_1$ and $Q_2$ in transcoder ($Q_1>Q_2$)

As shown in Figure 3.11, for $Q_1<Q_2$, one or more reconstruction points of the first quantizer will be mapped to a reconstruction point in the second quantizer. For $Q_1>Q_2$, as shown in Figure 3.12, not all reconstruction points of the second quantizer will be mapped by the first quantizer. In both figures, all uncompressed input in Vornoi region $j$ bounded by $S_{j-1}$ and $S_j$ will be mapped to the reconstruction point $T_j$. Note that $S_j$ are the break points of the first quantizer, and $T_j$ are reconstruction points of the second quantizer. Thus, $S_j$ are function of $Q_1$, and $T_j$ are function of $Q_2$. However, depending on the relative size of $Q_1$ and $Q_2$, $S_j$-$S_{j-1}$ may not be $Q_1$, but an integer multiple of $Q_1$. Similarly, $T_j$-$T_{j-1}$ may not be $Q_2$, but an integer multiple of $Q_2$. As discussed in Section 2.3.2, successive quantization may cause requantization error. This effect will be dealt with in the rate/distortion model developed in this section.

Note, when $Q_1=Q_2=Q$, $S_j=B_{1,j}=B_{2,j}=(\gamma+j)Q$ and $T_j=R_{1,j}=R_{2,j}$, i.e., all break points and reconstruction points of the two quantizers are aligned. In this case, no quantization error will be introduced by the second quantizer.

The derivation used in this section follows the same approach as the one taken in Section 3.1. The transcoding rate model is first developed in 3.2.1. It is then followed by

the transcoding distortion model developed in 3.2.2. Finally, a new method to estimate the Laplacian parameter for quantized input is presented in 3.2.3.

### 3.2.1. Laplacian Distortion Model for MPEG Video Transcoding

The derivation of the Laplacian distortion model for video transcoding is very similar to the one for the encoding. The main difference is that there is no fixed relationship between the breakpoint of the first quantizer and the reconstruction point of the second quantizer.

As shown in Figure 3.11 and Figure 3.12, when the original input ($x$) is requantized with $Q_2$, all input within Vornoi region $j$ ($S_j < x \leq S_{j+1}$) will be reconstructed at $T_j$. Thus, the expected distortion ($D_j$) of Vornoi region $j$ at the output of the transcoder is

$$\begin{aligned} D_j &= \int_{S_j}^{S_{j+1}} (x - T_j)^2 f(x)\,dx = \int_{S_j}^{S_{j+1}} (x - T_j)^2 \left(\frac{\alpha}{2} e^{-\alpha x}\right) dx \\ &= \left\{ -\frac{1}{2} e^{-\alpha x} \left[ x^2 + \left(\frac{2}{\alpha} - 2T_j\right) x + \left(\frac{2}{\alpha^2} - \frac{2}{\alpha} T_j + T_j^2\right) \right] \right\} \Bigg|_{x=S_j}^{S_{j+1}} \end{aligned} \tag{22}$$

Denote $G(x,t) = -\frac{1}{2} e^{-\alpha x} \left[ x^2 + \left(\frac{2}{\alpha} - 2t\right) x + \left(\frac{2}{\alpha^2} - \frac{2}{\alpha} t + t^2\right) \right]$, then

$$D_j = G(S_{j+1}, T_j) - G(S_j, T_j) \tag{23}$$

Hence, the expected distortion at the output of the transcoder is

$$
\begin{aligned}
&MSE(Q_1,Q_2 \mid \alpha) = 2\left[ D_0^+ + \sum_{j=1}^{\infty} D_j \right] \\
&= 2\left\{ \left[ G(S_1,T_0) - G(S_0,T_0) \right] + \left[ G(S_2,T_1) - G(S_1,T_1) \right] + \left[ G(S_3,T_2) - G(S_2,T_2) \right] + \cdots \right\} \\
&= 2\left\{ -G(S_0,T_0) + \sum_{j=1}^{\infty} \left[ G(S_j,T_{j-1}) - G(S_j,T_j) \right] \right\} \\
&= \frac{2}{\alpha^2} + \sum_{j=1}^{\infty} e^{-\alpha S_j} \left[ (T_j^2 - T_{j-1}^2) - 2\left( S_j + \frac{1}{\alpha} \right)(T_j - T_{j-1}) \right]
\end{aligned}
\tag{24}
$$

where $D_0^+$ is the distortion of the positive side of Vornoi region 0, i.e., $D_0^+ = G(S_1,T_0) - G(0,T_0)$.

Note, although $Q_1$ and $Q_2$ do not appear in (24) directly, this equation is still a function of $Q_1$ and $Q_2$ because $S_j$ and $T_j$ depend on $Q_1$ and $Q_2$ respectively. Also, since there is no fixed relationship between $S_j$ and $T_j$, unlike in Section 3.1.1, there is no closed-form solution for (24). As a result, the *MSE* function is evaluated numerically in the simulations.

As an implementation note, even though infinite summation is not feasible in practice, equation (24) can be evaluated with very little effort. Since each summation term in (24) is multiplied by $e^{-\alpha S_j}$, as $S_j$ increases, the summation term decreases exponentially, and the summation converges pretty rapidly. Convergence can be achieved in usually 5~10 summation terms. Thus, the computational complexity of (24) is not great compared to all other work done in compression.

As a final remark, although the MSE equations (17) (for encoding) and (24) (for transcoding) look quite different, it can be shown that when $Q_1=Q_2=Q$, fixed relationships between $S_j$ and $T_j$ can be found. By substituting $S_j = (\gamma + j)Q$ and $T_j = (\rho + j)Q$ into (24), equation (24) will degenerate to (17).

### 3.2.2. Laplacian Rate Model for MPEG-4 Video Transcoding

Like the previous section, the derivation of the Laplacian rate model for video transcoding is very similar to the one for the encoding. The difference is that there is no fixed relationship between the breakpoint of the first quantizer and the reconstruction point of the second quantizer. It also uses the MPEG-4 VLC Table model developed in Section 3.1.2 to estimate the number of bits per DCT coefficient.

Following the same approach as the one taken in Section 3.1.3, by passing the Laplacian distributed input through the generalized quantizer (see Figure 3.1), the output level ($k$) has the probability density function $f_L(k)$ as

- For $k=0$: $f_L(k=0\,|\,\alpha)=2\int_{S_0=0}^{S_1}\frac{\alpha}{2}e^{-\alpha x}dx=1-e^{-\alpha S_1}$
- For $k\neq 0$: $f_L(|k|\neq 0\,|\,\alpha)=2\int_{S_k}^{S_{k+1}}\frac{\alpha}{2}e^{-\alpha x}dx=e^{-\alpha S_k}-e^{-\alpha S_{k+1}}$

With the knowledge of the PDF of each quantized DCT level, and the bit cost of each level (18), the expected number of bits for each DCT coefficient can be obtained as

$$\begin{aligned}
Bit(Q_1,Q_2\,|\,\alpha) &= z\left(1-e^{-\alpha S_1}\right)+\sum_{k=1}^{M}(ak+b)\left[e^{-\alpha S_k}-e^{-\alpha S_{k+1}}\right]+\sum_{k=M+1}^{\infty}c\left[e^{-\alpha S_k}-e^{-\alpha S_{k+1}}\right] \\
&= \left[z-ze^{-\alpha S_1}\right]+\left\{\begin{array}{l}(a+b)e^{-\alpha S_1}-(aM+b)e^{-\alpha S_{M+1}}+\\ \sum_{k=2}^{M}e^{-\alpha S_k}\left[(ak+b)-(a(k-1)+b)\right]\end{array}\right\}+ce^{-\alpha S_{M+1}} \\
&= z+e^{-\alpha S_1}(a+b-z)-e^{-\alpha S_{M+1}}(aM+b-c)+a\sum_{k=2}^{M}e^{-\alpha S_k}
\end{aligned} \tag{25}$$

Since the value of $S_j$ depends on the relationship between $Q_1$ and $Q_2$, the summation cannot be simplified further. However, from Table 3.1, the values of $M$ are small, and the computational complexity of (25) is not great. Also, when $Q_1$=$Q_2$, it can be easily shown that (25) will degenerate to (20).

### 3.2.3. Laplacian Parameter Estimation for MPEG-4 Video Transcoding

For transcoder, one way to estimate the Laplacian parameter was proposed in [79]. However, it was developed for the MPEG-2 intra and inter quantizers. In MPEG-4, a total of four different quantizers can be used (intra/inter for H.261/MPEG-2 quantizers). Although all these quantizers are uniform quantizers, each quantizer has its unique way to set the dead-zone. Instead of developing four different ways to estimate the Laplacian parameter, a general approach is preferable since it can be applied to any arbitrary quantizers that might have been or will be developed.

Following the same approach as the one taken in [79], this section presents a method to estimate the Laplacian parameter when the input is quantized by a generalized uniform quantizer (see Figure 3.1). In other words, the solution presented in this section is a generalization of the solution provided in [79].

By passing the two-sided Laplacian distributed input through the generalized quantizer, the absolute output level ($k$) has the probability density function $f_L(|k|)$ as

$$f_L(|k|) = \begin{cases} 2\int_0^{\gamma Q} \frac{\alpha}{2} e^{-\alpha x} dx = 1 - e^{-\alpha \gamma Q} & k = 0 \\ 2\int_{(\gamma+|k|-1)Q}^{(\gamma+|k|)Q} \frac{\alpha}{2} e^{-\alpha x} dx = e^{-\alpha(|k|-1+\gamma)Q}\left(1 - e^{-\alpha Q}\right) & |k| \neq 0 \end{cases} \tag{26}$$

Let $y$ be the non-zero reconstructed DCT coefficient of level $k$ at input of the transcoder. From Figure 3.1, $y$ can be expressed in terms of $k$ and $Q$ as

$$|y| = (\rho + |k| - 1)Q = (|k| - 1)Q + \rho Q \tag{27}$$

or

$$(|k| - 1)Q = |y| - \rho Q \tag{28}$$

Substitute (28) into (26):

$$f_L\left(|k|\neq 0\right)=e^{-\alpha(\gamma Q+|y|-\rho Q)}\left(1-e^{-\alpha Q}\right)=e^{-\alpha(|y|-(\rho-\gamma)Q)}\left(1-e^{-\alpha Q}\right) \tag{29}$$

Let $p(\mathbf{y}) = P(y_1, y_2, \ldots y_N)$ be the joint probability of $N$ observed values of $y$, and there are $N_0$ number of zeros inside these $N$ samples, then

$$\begin{aligned} p(y) &= \left(1-e^{-\alpha\gamma Q}\right)^{N_0}\prod_{n=1,y_n\neq 0}^{N}\left\{e^{-\alpha(|y_n|-(\rho-\gamma)Q)}\left(1-e^{-\alpha Q}\right)\right\} \\ &= \left(1-e^{-\alpha\gamma Q}\right)^{N_0}\left(1-e^{-\alpha Q}\right)^{N-N_0}\prod_{n=1,y_n\neq 0}^{N}e^{-\alpha(|y_n|-(\rho-\gamma)Q)} \end{aligned} \tag{30}$$

From maximum likelihood theory, the Laplacian parameter $\alpha$ should satisfy

$$\frac{\partial}{\partial\alpha}\ln p(y)=0 \tag{31}$$

Hence,

$$\begin{aligned} &\frac{\partial}{\partial\alpha}\ln\left\{\left(1-e^{-\alpha\gamma Q}\right)^{N_0}\left(1-e^{-\alpha Q}\right)^{N-N_0}\prod_{n=1,y_n\neq 0}^{N}e^{-\alpha(|y_n|-(\rho-\gamma)Q)}\right\} \\ &=\frac{\partial}{\partial\alpha}\left\{N_0\ln\left(1-e^{-\alpha\gamma Q}\right)+\left(N-N_0\right)\ln\left(1-e^{-\alpha Q}\right)-\sum_{n=1,y_n\neq 0}^{N}\alpha\left(|y_n|-(\rho-\gamma)Q\right)\right\} \\ &=N_0\gamma Q\frac{e^{-\alpha\gamma Q}}{1-e^{-\alpha\gamma Q}}+\left(N-N_0\right)Q\frac{e^{-\alpha Q}}{1-e^{-\alpha Q}}-\sum_{n=1,y_n\neq 0}^{N}\left(|y_n|-(\rho-\gamma)Q\right) \\ &=0 \end{aligned} \tag{32}$$

or

$$\begin{aligned} &N_0\gamma Q\left(1-e^{-\alpha Q}\right)+\left(N-N_0\right)Q\left(1-e^{-\alpha\gamma Q}\right)- \\ &\left(1-e^{-\alpha Q}\right)\left(1-e^{-\alpha\gamma Q}\right)\sum_{n=1,y_n\neq 0}^{N}\left(|y_n|-(\rho-\gamma)Q\right)=0 \end{aligned} \tag{33}$$

Let $z = e^{-\alpha Q}$, $M = N - N_0$, $A = \sum_{n=1, y_n \neq 0}^{N} \left( |y_n| - (\rho - \gamma)Q \right) = (N - N_0)(\gamma - \rho)Q$, $B = N_0 \gamma Q$,

and $C = (N - N_0)Q$. Then, (33) becomes

$$Bz^{\gamma}(1-z) + Cz(1-z^{\gamma}) - (1-z)(1-z^{\gamma})A = 0 \tag{34}$$

Regrouping $z$ and $z^{\gamma}$, (34) becomes

$$-(A+B+C)z^{\gamma+1} + (A+B)z^{\gamma} + (A+C)z - A = 0 \tag{35}$$

Note that since both $\alpha$ and $Q$ are positive, $0 < z = e^{-\alpha Q} < 1$. Thus, although there might be multiple solutions to (35), the valid solution of $z$ must be between 0 and 1. Once the zero is found, by using the definition of $z$, the estimated Laplacian parameter is

$$\alpha = -\frac{\ln z}{Q} \tag{36}$$

As mentioned in Section 3.1.1, in MPEG-4, when the H.263 quantizer is used, $\gamma$=1 for intra quantizer, and $\gamma$=5/4 for inter quantizer. For $\gamma$=1, (35) becomes a quadratic equation, and the zeros are at

$$z = \frac{(2A+B+C) \pm \sqrt{(2A+B+C)^2 - 4A(A+B+C)}}{2(A+B+C)} \tag{37}$$

However, for $\gamma$=5/4, due to the non-integer power, there is no closed-form solution for (35). Thus, numerical solution is used in simulations. Although there are many numerical methods to solve (35), binary search is used since the valid $z$ is bounded.

## 3.3. Chapter Summary and Remarks

In this chapter, the Laplacian rate-distortion models for MPEG-4 encoder and transcoder are introduced. These models are derived based on the assumption that DCT coefficients have Laplacian distributions. Since the model parameters (Laplacian parameter) are derived from the input DCT coefficients of the current frame, temporal stationarity of the sequence is not assumed. Also, since generalized quantizer is used, the quantizer dead-zone is considered in the derivation. As demonstrated in the results, the models predict both the rates and the distortions quite accurately. These R/D models are used by the rate-control algorithms presented in the following chapters to help determining the QP(s) based on the specific rate/distortion constraints.

As a final remark, the rate/distortion models developed in this chapter are based on assuming the input having Laplacian distributions. As discussed in the beginning of this chapter, other distributions could also be used to model the DCT coefficients. If one decides to use another distribution, the same approach taken in this chapter can still be used to develop new rate/distortion models for other distributions. The Laplacian distribution is chosen because it is the most widely assumption used in the research. Also, the simple definition of the Laplacian PDF makes the analysis simple and clear. As shown in this chapter, close-form solution is obtained in many instances.

# Chapter 4

# Constant Bit Rate Control Algorithms

In the past, most rate control research emphasized constant bit rate (CBR) control. One reason for this is the requirement of transmitting video over fixed bandwidth communication channels. Having constant bitrate also facilitates the design of hardware video decoders since the maximum buffer size can be determined from the bitrate at design time.

The CBR constraint is achieved by having a fixed-size buffer. Buffer is used as a temporary storage of the encoded bitstream. Depending on the target bitrate and the buffer size, a fixed number of bits are taken out of the buffer for every fixed time interval. Thus, the buffer is used to regulate the maximum number of bits that can be transmitted in a fixed time interval. In CBR applications, one should always avoid buffer overflow. In most cases, avoiding buffer underflow is also desired.

In general, CBR control algorithms work according to the following three steps.

1. Allocate the target number of bits for each frame according to the target bitrate and current buffer status.
2. Compute the quantization parameter(s) (QPs) according to certain rate/distortion model. For frame-level quantization, only one QP is used for all macroblocks (MBs) within the frame. For macroblock-level quantization, QPs may vary from MB to MB.

3. After encoding the frame, the rate/distortion (R/D) model parameters are updated for future use.

In general, rate-control algorithms regulate rate/distortion by adjusting the QPs. In MPEG-2/4 and many other video compression standards, QPs can be changed at both the frame and the macroblock level. If a rate control strategy calls for frame-level QP control, the same QP must be used for all macroblocks in the same frame. Since there are only 31 possible choices for QP (for 5-bit precision quantizer, QP≠0), it is relatively simple to compute a QP that is optimum for some criteria. However, this may result in deviations from the desired target since only a small number of choices are available for each frame. In contrast, if a rate control strategy calls for macroblock-level QP control, different macroblocks within the same frame are allowed to use different QPs. Since there are many more choices of QPs per frame, the desired target may be more closely matched; however, the selection of macroblock-level QPs is more complex to compute than the selection of frame-level QP. In addition, extra overhead will be required to transmit the QP changes. This may lead to a larger bitstream and lower compression. Note that for both rate control strategies, the complexity at the decoder side remains the same.

In this chapter, two CBR algorithms are proposed that use different R/D models. In Section 4.1, a CBR algorithm for MPEG-4 is presented that uses the Laplacian rate model developed in Section 3.1.3. It is referred to as LPECBR (Laplacian Parameter Estimation CBR) algorithm [108]. Like Q2 [7], it controls the QP at the frame-level. Section 4.2 presents another CBR algorithm for MPEG-2 that improves the subjective quality measured in Watson's digital video quality (DVQ) [3][4] metric. It is referred to

as SQBCBR (Subjective-Quality-Based CBR) algorithm [109]. Like TM5 (Test Model 5) [8], it controls the QPs at the macroblock level.

## 4.1. MPEG-4 Laplacian Parameter Estimation CBR (LPECBR) Control Algorithm

In general, for intra frames, the DC DCT coefficients can be modeled as having uniform distributions [9], while the AC DCT coefficients can be modeled as having Laplacian distributions [9][98]. Since the bits for each frame are generated by entropy coding the quantized DCT coefficients, the number of bits used to code the frame is affected by the distribution of the DCT coefficients as well.

### 4.1.1. The LPECBR Algorithm

The LPECBR algorithm [108] is designed for MPEG-4, and is based on the Q2 algorithm. It uses the same target-bit allocation algorithm as used in the Q2 algorithm. On the other hand, it uses the Laplacian rate model to select the best quantization parameter for the given bit budget. It aims at meeting the target bit more accurately than Q2.

The best QP is the QP that generates the number of bits closest to the target, i.e.,

$$QP_{Selected} = \arg\min_{Q} \left| Bit_{Target} - \sum_{i=1}^{8}\sum_{j=1}^{8} Bit\left(Q \mid \alpha_{i,j}\right) \right| \tag{38}$$

where $Bit(Q|\alpha_{i,j})$ defined in (20) is the expected number of bits when DCT coefficients of frequency $i,j$ with Laplacian parameter $\alpha_{i,j}$ are quantized with $Q$.

### 4.1.2. Simulation Results and Discussions

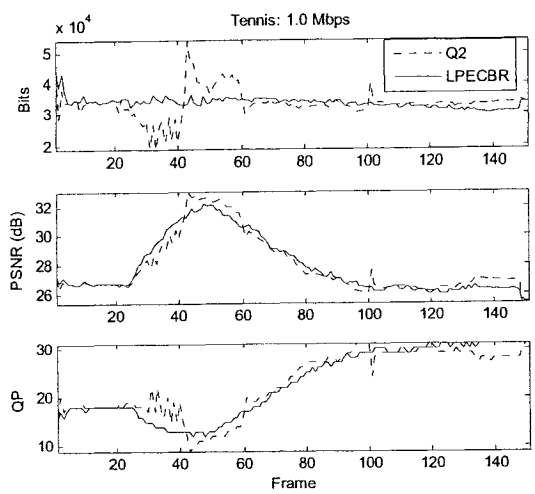
Two test sequences (table tennis and football, 150 frames each) are MPEG-4 encoded (with pure I-frames) using both the proposed and the Q2 algorithms at different bitrates. The simulation results are shown in Table 4.1 and Figure 4.1.

Note that for all simulations in this thesis, to ensure the correctness in coding the encoders (MPEG-2 and MPEG-4) and the reference algorithms (Q2 and TM5), the encoders and the reference algorithms are complied based on the original ISO (International Organization for Standardization) reference software source code (written in C language). Furthermore, to ensure the program is free of possible errors caused by compilers, the same source code is complied using three different compilers under different operating systems (Microsoft Visual C++ 6.0 for Windows, Microsoft Visual C++.NET 1.0 for Windows, and GNU G++ 3.3.2 for UNIX). These three encoder versions are tested by encoding the same video sequence (table tennis) with the same GOP structure (IBBP with intra period of 15 frames) at the same bitrate (1000 kbps). In each test, two outputs are generated: the decompressed video in raw form and the encoded bitstream. The encoded bitstream is then decompressed with the standard decoder to produce a second version of decompressed video in raw form. Finally, these raw videos (six versions) are compared with each others. This test procedure ensures all encoder versions give the same behaviour when they are complied by different compliers and run under different operating systems. It also increases the confidence level that the implementations of the proposed algorithms are based on a trustworthy encoder, and the simulation results obtained in this thesis are valid.

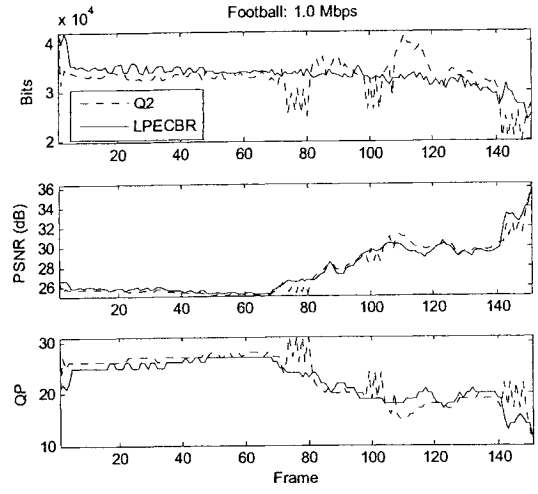
As shown in Table 4.1 and Figure 4.1, the LPECBR algorithm meets the target rate more accurately than Q2, and this is essential for CBR application. This is especially true for the 8 Mbps cases. Another important feature of the LPECBR algorithm is that it has smaller rate variance than Q2. Having smaller rate variance means that the buffer size can be reduced. Smaller buffer translates to lower transmission delay and lower cost for the digital video receiver. Note that the only exception where LPECBR has larger rate variance than Q2 is in the table tennis 8 Mbps case. For this case, although LPECBR has the larger rate variance than Q2, it meets the target bitrate more accurately than Q2. As shown in Figure 4.1.f, the Q2 is always under-spending the bits while the rate of the LPECBR is oscillating about the target rate, which increases the rate variance. As a final remark, the LPECBR produces a similar PSNR variance as Q2 does. Although the PSNR variation is not considered by both algorithms, it will be considered in the next chapter.

Table 4.1: Rate/PSNR comparison between Q2 and the LPECBR algorithm at different bitrates

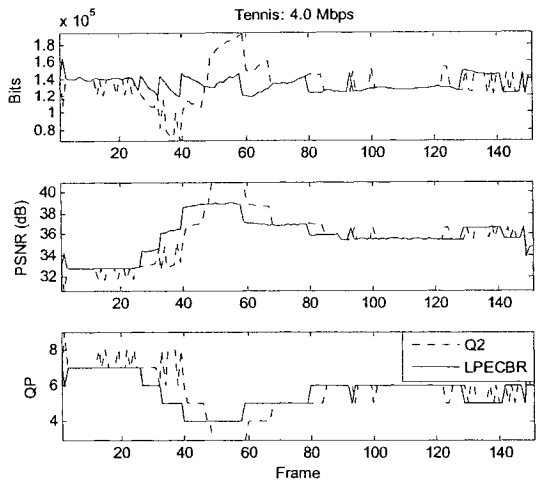
| | Table Tennis | | Football | |
|---|---|---|---|---|
| | Q2 | LPECBR | Q2 | LPECBR |
| Target Rate: 1000 kbps | | | | |
| Rate (kbps) | 1000.66 | 1007.85 | 980.08 | 999.80 |
| Rate Variance ($x10^6$) | 20.82 | 4.30 | 14.01 | 6.13 |
| PSNR (dB) | 27.82 | 27.84 | 27.60 | 27.70 |
| PSNR Variance | 3.97 | 3.82 | 5.84 | 5.52 |
| Target Rate: 4000 kbps | | | | |
| Rate (kbps) | 4007.68 | 4000.05 | 3930.07 | 4005.04 |
| Rate Variance ($x10^6$) | 427.41 | 73.52 | 355.33 | 68.97 |
| PSNR (dB) | 35.75 | 35.80 | 35.42 | 35.50 |
| PSNR Variance | 5.27 | 3.09 | 7.46 | 7.72 |
| Target Rate: 8000 kbps | | | | |
| Rate (kbps) | 6871.65 | 7996.63 | 6931.05 | 8017.74 |
| Rate Variance ($x10^6$) | 856.33 | 1063.87 | 2420.74 | 711.52 |
| PSNR (dB) | 40.59 | 42.25 | 40.27 | 41.41 |
| PSNR Variance | 0.25 | 3.46 | 1.11 | 5.63 |



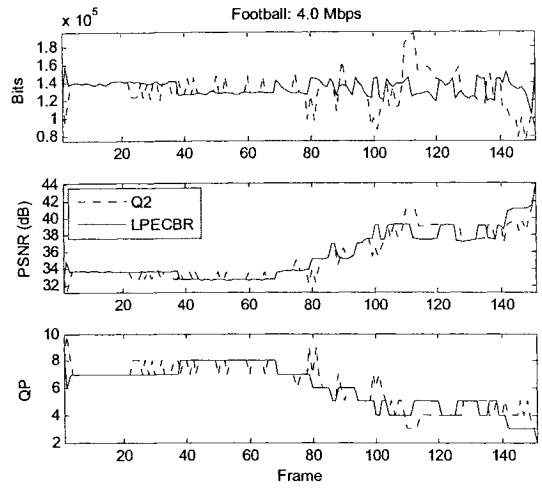

a) Table Tennis at 1.0 Mbps



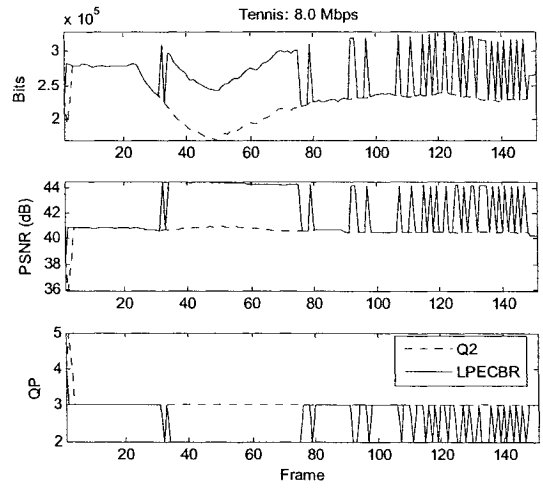

b) Football at 1.0 Mbps



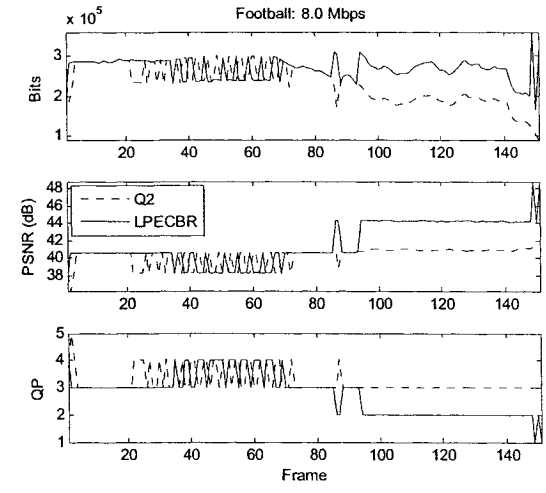

c) Table Tennis at 4.0 Mbps




d) Football at 4.0 Mbps




e) Table Tennis at 8.0 Mbps




f) Football at 8.0 Mbps

Figure 4.1: Rate, distortion, and quantization parameters of Q2 (dashed curve) and LPECBR (solid curve) algorithms of table tennis and football at different bitrates.

## 4.2. MPEG-2 Subjective-Quality-Based CBR (SQBCBR) Control Algorithm

In the LPECBR algorithm (for MPEG-4) presented in the previous section, the only concern is to produce a CBR bitstream. The quality of the bitstream has not been taken into account. In this section, a MPEG-2 Subjective-Quality Based CBR (SQBCBR) [109] control algorithm is proposed that attempts to improve the subjective quality while keeping the CBR constraint.

In MPEG-2, the recommended rate control algorithm is TM5 [8]. Since human beings are the final receivers of the video, it would be ideal to obtain the best perceptual video quality for the same bitrate. Unfortunately, TM5 does not take the perceptual quality into account. In this section, Watson's DVQ [3][4] metric is used by the SQBCBR algorithm to estimate the subjective quality.

### 4.2.1. SQBCBR and TM5

TM5 works in three steps: target-bit allocation, rate control, and adaptive quantization. The SQBCBR is built based on TM5 by changing the way target-bit is allocated for each frame. In TM5, the target bit ($T_{I/P/B}$) for I/P/B frame is allocated as follows.

$$T_{I/B/P} = \max\left\{\frac{R}{TB_{I/B/P}}, \frac{Bit\_rate}{8 \times Picture\_rate}\right\},\ X_{I/B/P} = S_{I/B/P} \cdot Q_{I/B/P}$$

$$TB_I = 1 + \frac{N_P}{X_I}\frac{X_P}{K_P} + \frac{N_B}{X_I}\frac{X_B}{K_B},\ TB_B = N_P + N_B\frac{K_P}{K_B}\frac{X_B}{X_P},\ TB_P = N_B + N_P\frac{K_B}{K_P}\frac{X_P}{X_B}$$

where

- $T$: Target bits allocated for next frame in the group of picture (GOP).
- $R$: Remaining bit assigned for this GOP.
- $X$: Global complexity measure
- $S$: Bit spent for encoding current frame.
- $Q$: Average q-factor of current frame
- $N_{P/B}$: Remaining number of B/P-frames in this GOP.
- $K_{P/B}$: Universal constant for the quantization matrix. For the default MPEG-2 quantization matrices, $K_P$=1.0 and $K_B$ = 1.4.

Note that in TM5, the target-bit allocation uses average q-factor to compute the global complexity measure ($X$). Since the MSE is proportional to the square of the quantization step size (proportional to the q-factor), higher q-factor means worse quality. Since the result of Watson's DVQ is an error metric, and higher error value means worse quality, the SQBCBR algorithm integrates Watson's DVQ into TM5 by replacing the average q-factor ($Q_{I/B/P}$) with the Watson's DVQ error metric, i.e., $X_{I/B/P} = S_{I/B/P} \times DVQ$.

Besides replacing the average q-factor with Watson's DVQ, more modification is needed. In TM5, the global complexity measure ($X$) for the next frame of the same type is updated based on the current frame, and the encoding order of the frames is independent from the displaying order of the frames. On the other hand, Watson's DVQ uses temporal-filtering, which depends on the display order of the frames. Since the encoding order ($I_0P_3B_1B_2P_6B_4B_5$...) is not the same as displaying order ($I_0B_1B_2P_3B_4B_5P_6$...), the DVQ calculations of I- and P-frames are delayed. Note that no extra frame buffering is needed because previous frames are already buffered by the ME/MC (motion

estimation/motion compensation) component of the encoder. Thus, the frame information can be reused by the DVQ.

### 4.2.2. Simulation Results and Discussions

Since the universal constants ($K_{P/B}$) in TM5 are scaling factors of the $X_{P/B}$, and the DVQ is used to compute $X$ instead of average q-factor, $K_{P/B}$ may also need to be changed. To find the optimum $K_{P/B}$ for the SQBCBR algorithm, simulations are performed.

As in TM5, $K_P$ is kept at 1.0. By varying the value of $K_B$, the effect of $K_B$ on the overall quality is investigated. The goal of the simulation is to find the optimum $K_B$ such that the quality of the encoded sequence using DVQ is better than the one using TM5. Note that since DVQ tries to estimate the perceptual quality, better DVQ does not imply better PSNR, and better PSNR does not imply better DVQ.

Three test sequences (table tennis, football, and Coke) are used in the simulation. Each sequence has 180 frames and is encoded at 2 Mbps using I/B/P frames with GOP of 15 frames with 3 frames between the I- and P-frames. Each sequence is encoded using two MPEG-2 encoders: one uses TM5 and the other uses the SQBCBR algorithm. The simulation results are compared in terms of both PSNR and Watson's DVQ. The results are shown in Figure 4.2 to Figure 4.7 and summarized in Table 4.2. Note that the best $K_B$'s for all test sequences are around 1.0.

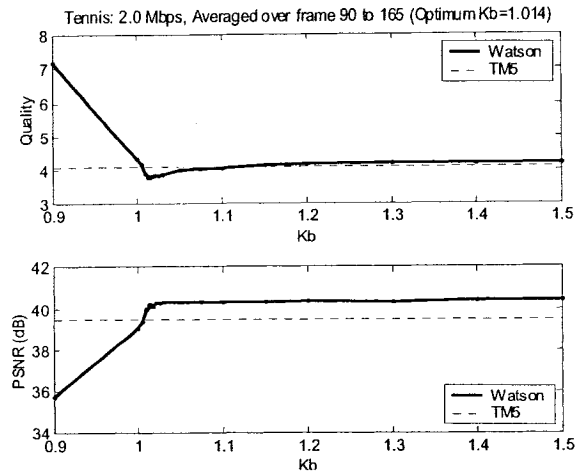



Figure 4.2: Effect of $K_B$ on Table Tennis [109]

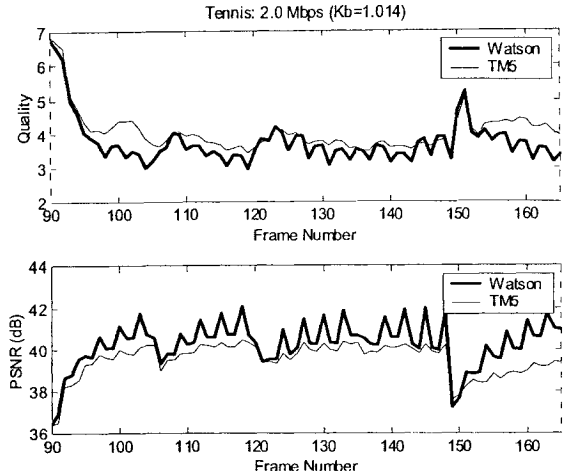



Figure 4.3: Watson vs. TM5 (Table Tennis) [109]

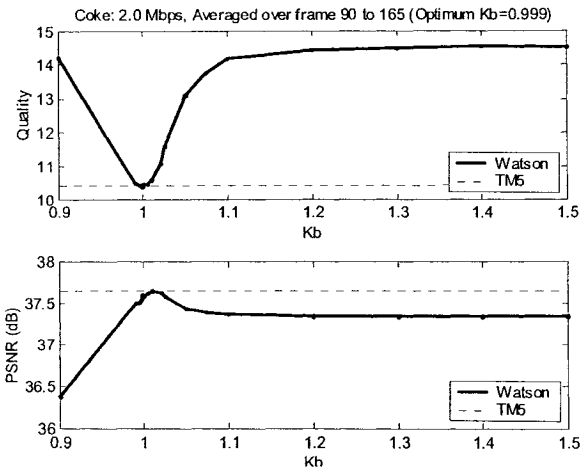



Figure 4.4: Effect of $K_B$ on Coke [109]

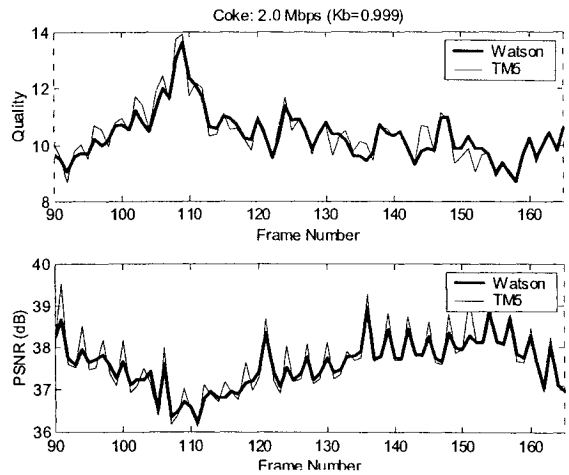



Figure 4.5: Watson vs. TM5 (Coke) [109]

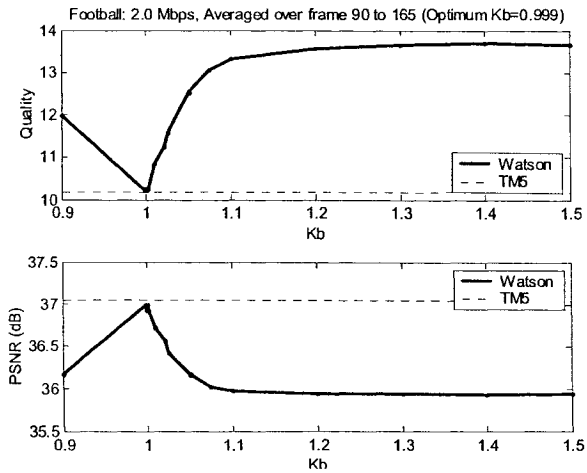



Figure 4.6: Effect of $K_B$ (Football) [109]

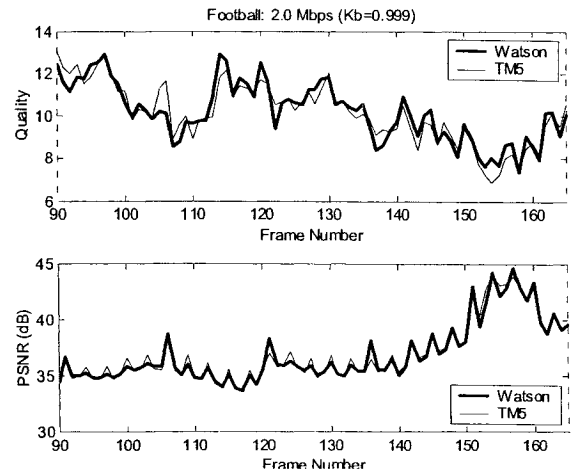



Figure 4.7: Watson vs. TM5 (Football) [109]

Table 4.2: TM5 vs. SQBCBR rate control algorithm [109]

| Sequence | | Table Tennis | Coke.com | Football |
|---|---|---|---|---|
| Best $K_B$ | | 1.014 | 0.999 | 1.001 |
| MPEG-2 Encoder using the TM5 rate-control algorithm | DVQ | 4.075 | 10.429 | 10.159 |
| | PSNR (dB) | 39.517 | 37.658 | 37.005 |
| | Bit Spent | 5018757 | 5039803 | 5045350 |
| MPEG-2 Encoder using the SQBCBR rate-control algorithm | DVQ | 3.773 | 10.387 | 10.207 |
| | PSNR (dB) | 40.264 | 37.605 | 36.953 |
| | Bit Spent | 4971048 | 5031864 | 5033492 |
| Improvement of the SQBCBR algorithm over TM5 | DVQ | 7.41 % | 0.40 % | - 0.47 % |
| | PSNR (dB) | 1.89 % | - 0.14 % | - 0.14 % |
| | Bit Spent | 0.95 % | 0.15 % | 0.24 % |

As shown in Table 4.2, for sequence table tennis, the SQBCBR algorithm performs significantly better than TM5, both in terms of the DVQ (lower is better) and the PSNR (higher is better). For sequence Coke, the SQBCBR algorithm performs slightly better than TM5 in DVQ, but slightly worse in PSNR. For sequence football, the SQBCBR algorithm performs slightly worse than TM5 both in terms of the DVQ and PSNR.

These simulation results suggest that the SQBCBR algorithm performs significantly better than TM5 for low-motion video sequence (table tennis). For high-motion sequences (Coke and football), the SQBCBR algorithm performs about the same as TM5. This phenomenon may be attributed to the following factors:

- The DVQ computation of the I- and P-frames is delayed in the SQBCBR algorithm because DVQ relies on the displaying order rather than the encoding order. Thus, the SQBCBR algorithm reacts slower than TM5. This affects the performance of the SQBCBR algorithm in high-motion scenes.
- The DVQ uses several low-pass filters. The choice of the time constants for these filters may affect the performance of the SQBCBR algorithm for high motion scenes.

### 4.2.3. Issues with the SQBCBR algorithm

There are several issues with the SQBCBR algorithm:

- **Sensitivity to $K_B$**: The SQBCBR is sensitive to value of $K_B$. From the simulation results, different $K_B$ yields different quality. Different bitrates have different optimum $K_B$.
- **Sensitivity to High Spatial Activity**: The sequence football has worse DVQ than the TM5, while other sequences have better DVQ. Since football contains high spatial activities, while table tennis and Coke do not, this suggests that the DVQ might be over-sensitive to the high spatial activity.
- **Computational Complexity**: The DVQ adds extra computation complexity to the encoder, which increases encoding time per frame.
- **Non-Standard Objective Quality Metric**: The notion of objective quality is still new today. The VQEG has not decided the final version of the quality model in both phase I [66] and phase II [110] reports.

## 4.3. Chapter Summary

Two CBR algorithms are proposed in this chapter. The MPEG-4 LPECBR algorithm uses the Laplacian rate model to compute the frame-level QP for a given target bit. Simulation results suggest that it meets the target bitrate more accurately and has smaller rate variance than the Q2 algorithm. The MPEG-2 SQBCBR algorithm uses the Watson's DVQ metric to improve the subjective quality. It offers better DVQ quality than TM5 for low-motion sequences, and it offers about the same DVQ quality than TM5 for high-motion sequences. However, as mentioned in Section 4.2.3, since the DVQ metric is not a standard way to predict the subjective quality, this thesis still uses traditional PSNR as the measure of quality in the following chapters.

For both algorithms proposed in this chapter, the main concern is to meet the target bitrate with better quality (i.e., the average quality). The variation in quality is not considered. In the next chapter, several rate-control algorithms are proposed that attempt to control the quality as well as the bitrate.

# Chapter 5

# MPEG-4 Constant Quality Rate Control Algorithms

In the past, most video rate-control research emphasized constant bit rate (CBR) applications. The aim of these algorithms was to produce a bitstream that meets a certain bitrate over some period of time (buffer control), and to achieve quality as high as possible. Due to the rate constraint and the non-temporally-stationary nature of the video sequence, the encoder has to spend approximately the same number of bits (CBR) for both easy-to-code frames and for hard-to-code frames. This results in better quality for easy-to-code video sections, and worse for hard-to-code video sections. Thus, the quality of the compressed sequence changes over time, and cannot be guaranteed. This is not desirable. In this chapter, constant quality (CQ) rate control algorithms are studied.

As mentioned in Chapter 4, rate-control algorithms regulate rate/distortion by adjusting the quantization parameters (QPs). In MPEG-4 and many other video compression standards, QPs can be changed at both the frame and the macroblock (MB) level. Frame-level control of QP is relatively simple to compute since only one QP needs to be determined for each frame. However, since there are only a small number of choices of QP in each frame, frame-level QP control may result in larger deviations from the desired quality. In contrast, macroblock-level QP control can meet the desired quality more closely. However, because there are many more choices of QPs, macroblock-level QP control is more complex to compute than frame-level QP control. In addition, in order

to properly decode the bitstream, changes in the macroblock QPs are also needed to be transmitted. This may lead to a larger bitstream and lower compression.

The most straightforward CQ rate control algorithm is to apply a constant QP for the entire sequence. Unfortunately, because of the non-stationary nature of video sequences, constant QP does not, in general, generate a constant-quality bitstream.

In the past few years, several CQ algorithms ([5],[6],[101]-[107],[111],[112]) have been reported, in which different QP control strategies, rate-distortion models, and channel properties are used/assumed.

Most of these algorithms use frame-level QP control. Only [104] and [106] use macroblock-level QP control. Different rate-distortion models are also used, such as the MAD (mean absolute difference) quadratic model in [101], the MPEG-4 quadratic model in [104], the Bayesian estimation model in [105], and the high-resolution model in [106]. In [102] and [103], algorithms are designed for transmitting video over ATM/DiffServ (Asynchronous Transfer Mode/Differentiated Service) channels. Except in [102] and [106] which use VBR (variable bit rate) channels, a CBR channel is assumed in all other work.

To achieve CQ, different approaches/contexts are used/assumed in these algorithms. In [101], multi-pass quantization is used to find the best QP for the bit budget determined by the target-bit allocation algorithm. In [102], a lossy DiffServ channel is assumed. Constant quality is achieved by sending video packets using either low-loss premium packets or best-effort packets depending on the perceptual importance of each MB. In [103], quality is controlled by using the minimum-maximum distortion (MMAX) criterion over an ATM channel with token bucket (TB) traffic shaping. Two algorithms

are proposed in [104] for base layer and enhanced layer control of MPEG-4 fine-grain scalability (FGS) video. With the help of enhanced layers, CQ is achieved within each GOV (group of video object plane), but not across the GOVs. In [105], instead of using the DCT (discrete cosine transform), a wavelet transform is used. QPs are adjusted based on the mean absolute wavelet coefficients. In [106], a controllable VBR (CVBR) channel is considered. A weighting factor is used to balance the buffer occupancy and/or the distortion variation. In [107], stochastic dynamic programming (SDP) is used to control the buffer status, distortion, and distortion variability (absolute distortion difference between adjacent frames). In [5] and [6], the quality is controlled indirectly through the target bit allocation using MAD (in [5]) or number of zeros (in [6]) to compute the target bits. Both [111] and [112] are intended for offline use. The rate/distortion information is computed offline prior to the actual encoding. The work in [111] is based on spectral FGS (SFGS), which is modified from MPEG-4 FGS. Dynamic programming (DP) is used to allocate bit/distortion for different frames and layers. For [112], the input sequence is first segmented into shots based on scene activities. Rate-distortion optimization is then applied to compute the best QP for each shot. Note that bidirectional coding is not allowed in [112].

While the term CQ is used in many of the reviewed work, until recently, there was no clear measure of how *constant* the quality is, and it is not possible to quantify the performance of these algorithms. In this chapter, quality is measured in PSNR (peak signal-to-noise ratio). *CQ* is measured by the variance of the PSNR over time. This CQ measure is also used in [5] and [6] (PSNR standard deviation). A good CQ control

algorithm should have low PSNR variance (ideally zero) while having high average PSNR.

Six CQ control algorithms are proposed in this chapter. Unlike all previously reported CQ algorithms that control the quality indirectly through the target-bit allocation, the proposed CQ algorithms control the quality directly through quality matching (QM)/bit matching (BM) algorithms that are presented in Section 5.1. The QM/BM algorithms compute the best QP(s) for a given target-quality/bit-budget. They are used by all CQ algorithms presented in this chapter. The first two CQ algorithms are presented in Section 5.2. They are designed for VBR applications with a fixed target quality. The other four CQ algorithms are for CBR applications. Two of them presented in Section 5.3 use a fixed target quality, and the other two presented in Section 5.4 adjust the target quality dynamically. Within each group of two CQ algorithms, one of them uses frame-level QP control, and the other uses MB level QP control. Section 5.5 summarizes this chapter.

## 5.1. Quality Matching and Bit Matching Algorithms

This section presents quality matching and bit matching algorithms. They are used to find the QP(s) such that the resulting compressed frame best matches a given target quality or bit budget. Two types of QM/BM algorithms are presented in this section. The Frame-level Laplacian QM/BM (FLQM/FLBM) [99] algorithms are presented in Section 5.1.1 and 5.1.2. They use the MPEG-4 Laplacian distortion/rate model for encoders developed in Section 3.1 to find the best QP for the entire frame. The Macroblock-level Viterbi QM/BM (MVQM/MVBM) [99] algorithms are presented in Section 5.1.3 and

5.1.4. Instead of using the Laplacian distortion model, they use real MSE (mean squared error) data and the Viterbi algorithm to find QPs for each macroblock.

### 5.1.1. Frame-Level Laplacian Quality Matching (FLQM) Algorithm

The FLQM algorithm is used to determine the QP that best matches the given target quality ($MSE_{Target}$):

**Step 1**: Use (21) to estimate the Laplacian parameters ($\alpha_{i,j}$) of the current frame, where $\alpha_{i,j}$ is the estimated Laplacian parameter of DCT frequency $(i,j)$.

**Step 2**: The selected QP is the QP with $MSE$ (defined in (17)) that is closest to the target MSE, i.e.,

$$QP_{Selected} = \arg\min_{QP} \left| MSE_{Target} - \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8} MSE\left(QP \mid \alpha_{i,j}\right) \right| \tag{39}$$

where $QP$= 2, 4, 6... 62.

In practice, some frames/DCT coefficients will not follow the Laplacian distribution. A correction factor ($\delta_{MSE}$) is added to Step 2 to compensate for any deviation. The correction factor is defined as the difference between the analytical MSE and the real MSE of the previously encoded frame of the same type. Thus, (39) becomes

$$\begin{aligned} QP_{Selected} &= \arg\min_{QP} \left| MSE_{Target} - \left( \delta_{MSE} + \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8} MSE\left(QP \mid \alpha_{i,j}\right) \right) \right| \\ \delta_{MSE} &= ActualMSE_{Previous} - \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8} MSE\left(QP_{Previous} \mid \alpha\_Previous_{i,j}\right) \end{aligned} \tag{40}$$

where $ActualMSE_{Previous}$ , $QP_{Previous}$ and $\alpha\_Previous$ are the actual MSE, QP, and Laplacian parameters of the previously coded frame of the same type.

### 5.1.2. Frame-level Laplacian Bit Matching (FLBM) Algorithm

Similar to the FLQM, the FLBM algorithm is used to determine the QP that best matches the given bit budget ($Bit_{Target}$):

**Step 1**: Use (21) to estimate the Laplacian parameters ($\alpha_{i,j}$) of the current frame, where $\alpha_{i,j}$ is the estimated Laplacian parameter of DCT frequency $(i,j)$.

**Step 2**: The selected QP is the QP with the estimated number of bits that is closest to the target bits, i.e.,

$$QP_{Selected} = \arg\min_{QP} \left| Bit_{Target} - \left( \delta_{Bit} + \sum_{i=1}^{8}\sum_{j=1}^{8} Bit\left(QP \mid \alpha_{i,j}\right) \right) \right|$$
$$\delta_{Bit} = ActualBit_{Previous} - \sum_{i=1}^{8}\sum_{j=1}^{8} Bit\left(QP_{Previous} \mid \alpha\_Previous_{i,j}\right) \tag{41}$$

where $QP$ = 2, 4, 6... 62, and $Bit(QP|\alpha_{i,j})$ is defined in (20), $ActualBit_{Previous}$ , $QP_{Previous}$ and $\alpha\_Previous$ are the actual number of bits, QP, and Laplacian parameters of the previously coded frame of the same type.

### 5.1.3. Macroblock-level Viterbi Quality Matching (MVQM) Algorithm

In order to meet the target distortion more accurately, a macroblock level quality control algorithm is derived to solve the following problem:

$$\arg\min_{QP_1 \ldots QP_M} \left| MSE_{Target} - \frac{1}{M}\sum_{i=1}^{M} MSE_i\left(QP_i\right) \right| \tag{42}$$

where $MSE_{Target}$ is the target MSE of the current frame, $QP_i$ is the quantization parameter for macroblock $i$, $M$ is the number of macroblocks, and $MSE_i(QP_i)$ is the MSE of macroblock $i$ when quantized with $QP_i$. Note that (42) can be reformulated as

$$\arg\min_{QP_1 \ldots QP_M} \left| \sum_{i=1}^{M} \left[ T_i - MSE_i\left(QP_i\right) \right] \right| \tag{43}$$

where $T_i$ is the target MSE of macroblock $i$, and

$$MSE_{Target} = \frac{1}{M}\sum_{i=1}^{M} T_i \tag{44}$$

Unlike in MPEG-2 where the standard does not put any limitation on the transition between $QP_i$ and $QP_{i+1}$, MPEG-4 allows only limited changes (referred to as *dquant* for I-/P- VOPs (video object planes), and *dbquant* for B-VOPs) between $QP_i$ and $QP_{i+1}$. For I- and P-VOPs, *dquant* can have values of -2, -1, 0, 1, and 2, i.e., one can at most change the QP by ±2 from one MB to the next. Note that for P-VOP MBs with four motion vectors (Inter4V mode), no *dquant* is allowed, i.e., one cannot change the QP for Inter4V macroblocks. As for B-VOPs, *dbquant* can have value of -2, 0, and 2. Thus, fewer choices for QP are available. All these constraints are considered in the MVQM algorithm.

Since there are a maximum of 5 possible QPs for each MB in I/P VOPs, if exhaustive search is used, for $N$ macroblocks, there are approximately $31\times 5^{N-1}$ possible choices of QPs (paths). Similarly, there are maximum of 3 possible QPs per each MB, and about $31\times 3^{N-1}$ paths for B-VOPs. Since $N$ is usually in the order of hundreds or thousands, the complexity of an exhaustive search is clearly too large. The well-known Viterbi Algorithm (VA) [113][114] is used to reduce the search complexity as is detailed in the next section.

#### *5.1.3.1. Using the Viterbi Algorithm in MVQM*

The Viterbi algorithm is a dynamic programming algorithm. It can be used to find the most likely path in a finite-state trellis of memoryless noise. Its complexity grows linearly with the length of the trellis. It is often used to solve the minimum path problem

in a trellis, where the distance between each state is non-negative. Since each summation term in (43) could be either positive or negative, the VA cannot be used directly.

One way to address this problem is to alter (43) so that summation terms are non-negative, i.e., approximate (43) with

$$\arg\min_{QP_1 \ldots QP_M} \sum_{i=1}^{M} \left| T_i - MSE_i(QP_i) \right| \tag{45}$$

The VA can then be used to find the optimum solution for (45). Note that since (43) allows cancellation of positive and negative summation terms but (45) does not, the optimum solution given by (45) is suboptimum with respect to (43). Simulation results indicate that (45) provides reasonably good results (see Section 5.2.4) for the problems studied in this thesis.

Since the global target ($MSE_{Target}$) is used to generate macroblock targets ($T_i$), for any $MSE_{Target}$, there are a large number of ways to assign the $T_i$. With (45), the VA only gives the optimum solution for each set of assigned $T_i$. The choice of the $T_i$, of course, affects the global optimality. In the next section, a method for setting the $T_i$ is discussed.

#### *5.1.3.2. Setting the Macroblock Target MSE ($T_i$)*

Three trivial ways to set the macroblock target MSE ($T_i$) are:

- **Averaging function**: $T_i = MSE_{Target}$.
- **Normalized MSE from the optimal frame QP**:

  The optimum frame QP is the constant QP for all MBs resulting in the closest MSE to the target MSE, i.e.,

$$QP_{Optimum} = \arg\min_{QP} \left| MSE_{Target} - \frac{1}{M} \sum_{i=1}^{M} MSE_i(QP) \right| \tag{46}$$

Since the MSEs of all MBs and QPs must be known in order to use this algorithm, it is very easy to find the optimum frame QP. Set the $T_i$ to be the normalized MSE of the optimal frame QP path, i.e.,

$$T_i = \frac{MSE_i\left(QP_{Optimum}\right)}{\frac{1}{M}\sum_{i=1}^{M} MSE_i\left(QP_{Optimum}\right)} \times MSE_{Target} \tag{47}$$

- **Normalized MSE from the optimal frame QP±1**:

  Similar to the previous case, instead of normalizing the MSE of the optimal frame QP, normalize the MSE of the adjacent QP that encloses the target MSE. To be more specific, if the MSE of the optimal frame QP is smaller than target MSE, normalize the MSE of QP+1. Otherwise, normalize the MSE of QP-1.

Note that all three methods of setting the initial $T_i$ satisfy equation (44). Since these trivial ways to set the $T_i$ are simple to compute and the VA uses low computation as well, one may try all three, and then select the result that gives the best match.

Further refinement is possible. This is done by running the VA repeatedly with a refined set of $T_i$ in each iteration. For each iteration, the new set of $T_i$ is obtained by normalizing the selected path MSEs of the previous iteration. The MVQM algorithm is outlined as follows.

**Step 1:** Set the initial $T_i$ using one of the methods as described above.

**Step 2:** Run the VA to minimize (45).

**Step 3:** If the selected QPs are the same as those in the previous iteration, or if the difference of the resulting MSE between the current and the previous iteration is smaller than a threshold, stop the algorithm.

**Step 4:** Set the new $T_i$ to

$$T_i = \frac{MSE_i(QP_i)}{\frac{1}{M}\sum_{i=1}^{M} MSE_i(QP_i)} \times MSE_{Target} \tag{48}$$

where $QP_i$ is the QP of macroblock $i$ selected in **Step 2** of this iteration. Note that the new set of $T_i$ still averages to $MSE_{Target}$.

**Step 5:** Go to **Step 2.**

By running Steps 2-5 repeatedly, it is possible to narrow the gap between the selected path MSE and the target MSE. In the simulations, all three trivial initial settings of the $T_i$ are used. Out of the three selected paths, the one with the closest MSE to the target MSE is chosen to encode the frame. Although the MVQM algorithm gives a suboptimal solution to (42), simulations indicate that the algorithm meets the target MSE very well (See Section 5.2.4).

#### *5.1.3.3. Complexity issues to compute MSE for all macroblocks in MVQM*

The MVQM algorithm achieves the target quality by finding a path such that the difference between the overall MSE and the target MSE is minimal for a given set of $T_i$. Instead of using exhaustive search (about $31 \times 5^N$ additions), the VA (about $5 \times 31 \times N$ additions, for each time VA is run) is used to reduce the complexity of the search for $N$

macroblocks in the case of I/P frames. Simulations indicate that, in general, the MVQM converges in 4~6 VA iterations.

The main complexity of the MVQM algorithm is the computation of the MSEs of all macroblocks for all QPs. In the simulation, this information is obtained by doing the following 31 times: quantizing and de-quantizing the DCT coefficients, and computing the MSE of the entire frame. Since the DCT is an orthogonal transform, instead of computing the MSE in the pixel domain, the MSE is computed in the DCT domain. The computational complexity of each 8x8 block is modeled as follows.

- Quantization and de-quantization: 128 multiplications and 128 additions.
- Compute MSE: 64 multiplications and 64 additions.

By repeating this process 31 times, obtaining the complete MSE information of each 8x8 block would require 5952 multiplications and 5952 additions. This process is computationally expensive and time consuming. For applications that have limited computation power or a limited encoding time constraint, it may not be feasible to do this. Several alternatives may be used to reduce the complexity further by sacrificing the MSE accuracy.

The main idea behind the fast (and suboptimum) alternatives is to reduce the number of candidates (either the number of QPs or length of the trellis). As an example, instead of quantizing/de-quantizing 31 times, one may quantize fewer times and interpolate the missing MSE data points. If quantization is only performed for QP=1, 6, 11, 16, 21, 26 and 31, the computational complexity may be reduced by 4.4 times.

One may also completely avoid the computationally-intensive quantization by using the MSE model derived in (17) to estimate the MSE. Since Laplacian parameters

are estimated from the DCT coefficients of the same frequency, and there are only four luminance blocks in each macroblock, the Laplacian parameter (hence the MSE) may not be estimated correctly since only a few samples are used. Thus, instead of allowing QPs to change for each macroblock, one may decide to change QPs for each group of $k$ macroblocks ($k > 1$). By doing so, one can get a better estimate of the Laplacian parameter and the MSE by sacrificing the number of opportunities that QP may change.

As an example, for CIF (352x240) sequences, if one decides to estimate the Laplacian parameters for groups of 4x4 non-overlapped macroblocks, one only needs to estimate the MSEs of 24 groups. Within each group, the computational complexity to estimate the MSE is the same as the FLQM algorithm. Since the computational complexity of FLQM is small, this alternative would provide a significant complexity improvement compared to the original. As a final remark, as the number of MBs in each group increases, both the computational complexity and the MSE accuracy approach to the FLQM (all MBs belong to one group).

### 5.1.4. Macroblock-level Viterbi Bit Matching (MVBM) Algorithm

The MVBM algorithm is used to determine the QPs for all MBs that best match to a given bit budget. In MVQM, the actual MB MSEs are used to compute the best MB QPs. However, this is not the case in MVBM.

In MPEG-4, the actual number of bits spent depends on whether the quantized DCT coefficients are differentially coded or not. Since the decision of whether to use differential coding is made after the quantization, it is not possible to know the exact number of bits before quantization and before determining the QPs. As an alternative to

counting the actual bits, the MVBM uses the Laplacian rate model discussed in Section 3.1.3 to estimate the bits.

The macroblock QPs are determined by searching for the highest target quality ($PSNR_{Max}$) that matches closest to the bit budget ($Bit_{Target}$). For each target quality searched, MVQM is first used to determine the macroblock QPs. The average MB QP is then fed into (20) to estimate the bits required for coding that quality. In short, given a target PSNR, subroutine *Compute_QPs_and_Bits* is used to find the MB QPs and estimate the bits required.

[*MB_QPs*, *EstimatedBits*] = Compute_QPs_and_Bits (*TargetPSNR*)

Begin

*MB_QPs* = MVQM(*TargetPSNR*);

*EstimatedBits* = Estimated bits using (20) with input QP set to average MB_QPs;

End.

The MVBM uses a combination of linear search (used to find an appropriate target PSNR range for this $Bit_{Target}$) and binary search (to refine the target PSNR) as described below.

**Step 1**: *Linear Search:*

a. Algorithm Initialization:

Set $PSNR_{Low}$ to $PSNR_{Initial}$ for the first frame, and the PSNR of the previous frame for all other frames. Set $PSNR_{High}=PSNR_{Low}+\delta_{LS}$ dB. Then, set

$[QP_{PSNRLow}, Bit_{PSNRLow}]$ = Compute_QPs_and_Bits($PSNR_{Low}$), and

$[QP_{PSNRHigh}, Bit_{PSNRHigh}]$ = Compute_QPs_and_Bits($PSNR_{Low}$).

In the simulations, $\delta_{LS}$ = 1 dB.

b. Stopping condition:

If $|Bit_{PSNRLow}-Bit_{Target}|/Bit_{Target} < ThresholdRatio\%$ (close enough to the bit budget) or $|Bit_{PSNRLow}-Bit_{PSNRLowPreviouseIteration}| < ThresholdBit$ bits (no more improvement), stop the MVBM. The MB QPs corresponding to $PSNR_{Low}$ ($QP_{PSNRLow}$) are used to code the frame. In the simulation, *ThresholdRatio* is set to be 5, and *ThresholdBit* is set to be one bit.

c. Move to Refinement Stage:

If $Bit_{PSNRLow} < Bit_{Target} < Bit_{PSNRHigh}$, go to Step 2.

d. Initialization for the next iteration:

i. If $Bit_{PSNRLow} > Bit_{Target}$, set $PSNR_{High}=PSNR_{Low}$, $Bit_{PSNRHigh}=Bit_{PSNRLow}$, and $QP_{PSNRHigh}=QP_{PSNRLow}$. Then, decrease $PSNR_{Low}$ by $\delta$ dB, and set the new $QP_{PSNRLow}$ and $Bit_{PSNRLow}$ as

$[QP_{PSNRLow}, Bit_{PSNRLow}]$ = Compute_QPs_and_Bits($PSNR_{Low}$).

ii. Otherwise (i.e. $Bit_{PSNRHigh} < Bit_{Target}$), set $PSNR_{Low}=PSNR_{High}$, $Bit_{PSNRLow}=Bit_{PSNRHigh}$, and $QP_{PSNRLow}=QP_{PSNRHigh}$. Then, increase $PSNR_{High}$ by $\delta$ dB, and set the new $QP_{PSNRHigh}$ and $Bit_{PSNRHigh}$ as

$[QP_{PSNRHigh}, Bit_{PSNRHigh}]$ = Compute_QPs_and_Bits($PSNR_{High}$).

e. Go to Step 1.b.

**Step 2**: *Binary Search*:

Binary Search is used to refine the target quality (bounded by $PSNR_{Low}$ and $PSNR_{High}$) further. The search stops either when it reaches the maximum iteration $Iteration_{Max}$ or when the estimated bit is within *ThresholdRatio*% of the target bit. Again, subroutine *Compute_QPs_and_Bits* is used in Binary Search to compute the MB QPs and to estimate bits for the refined target PSNRs. In the simulation, $Iteration_{Max} = 10$. The MB QPs corresponding to $PSNR_{Max}$ ($QP_{PSNRMax}$) are used to code the frame.

The convergence rate of the MVBM algorithm is pretty fast, usually within 2 to 5 iterations. Thus, the computational complexity is not great.

### 5.1.5. Using Other Quality Measures in the QM/BM Algorithms

The quality metric used by FLQM, MBQM, and MVBM (implicitly used) is MSE (converted from PSNR). However, it does not mean that the algorithms will not work for other non-MSE based quality metrics.

As discussed in Section 2.2.2, MSE is a statistical quality measure. There also exist many other kinds of subjective quality predictors, such as the Watson's DVQ (digital video quality) metric shown in Section 2.2.3. In order to apply other kinds of quality metrics to the QM/BM algorithms presented in this section, the quality metric must have the additive property. This means that the total frame error must be the same as simply adding the errors of every DCT coefficients and of every macroblocks. Thus, the Watson's DVQ metric can be used as the quality measure in QM/BM algorithms if

the pooling method (final step) is linear, i.e., use simple addition to add all JNDs (just noticeable differences) together to produce the quality.

The additive property of the quality metric is essential since all proposed QM/BM algorithms rely on it. As an example, the MVQM needs the additive property to break down the global distortion target into macroblock distortion targets in order to find a best combination of macroblock QPs. It also needs the additive property to add all macroblock distortions together to obtain the frame distortion. Thus, the proposed QM/BM algorithms will work for any quality metric as long as it has the additive property.

This also means that any rate control algorithm that uses the proposed QM/BM algorithm can use non-MSE based quality measures. However, for all CQ rate-control algorithms presented in this and the next chapter, MSE is used as the measure of quality.

## 5.2. MPEG-4 Constant-Quality VBR Algorithms with Fixed Target Quality

In all CQ algorithms reviewed, a buffer constraint is applied. The advantage of having a constrained buffer is to avoid overflow and underflow, which is very important for CBR coding. However, because the constrained buffer limits the bit budget, it affects the quality as well. To have a constrained buffer, rate-control algorithms need to spend more bits (increase the quality) when the buffer occupancy is low, and to spend fewer bits (reduce the quality) when the buffer occupancy is high. This introduces variations in the output quality, which is undesirable for CQ coding. To avoid this problem, the CQ algorithms presented in this section are VBR algorithms. They assume that the buffer has

infinite size, i.e., they assume an unconstrained buffer. Note, if these algorithms are used in a limited buffer size scenario (CBR), neither buffer overflow nor buffer underflow will be protected. The only goal of these algorithms is to achieve constant quality.

It should be remarked that even if the buffer size is said to be *infinite*; the buffer size is always bounded by the size of one uncompressed frame. Although having such a large buffer size is not suitable for video streaming, this condition is still good for offline video storage and when the channel bandwidth is very high.

Two CQ VBR algorithms are proposed in this section. The FLCQ algorithm presented in Section 5.2.1 controls the quality at the frame level, while the MVCQ algorithm presented in Section 5.2.2 controls the quality at the macroblock level. Section 5.2.3 discusses the reference CQ VBR algorithms used to evaluate the performance of the proposed algorithms. Simulation results and discussions are presented in Section 5.2.4.

### 5.2.1. MPEG-4 Frame-level Laplacian Constant-Quality (FLCQ) VBR Algorithm with Fixed Target Quality

Since neither the bitrate nor the buffer occupancy is important, the FLCQ VBR algorithm is simply the FLQM algorithm derived in Section 5.1.1. For the given fixed target quality, this algorithm only needs to use the FLQM to determine the best frame QP for all MBs within the frame.

### 5.2.2. MPEG-4 Macroblock-level Viterbi Constant-Quality (MVCQ) VBR Algorithm with Fixed Target Quality

Similarly, the MVCQ VBR algorithm is simply the MVQM algorithm derived in Section 5.1.3. For the given fixed target quality, this algorithm only needs to use the MVQM to determine the best QPs for every macroblocks.

### 5.2.3. Reference Constant Quality Rate Control Algorithms

In order to evaluate the performance of the proposed CQ algorithms, two frame-level reference CQ algorithms are used. The first reference algorithm applies a constant QP for all macroblocks in all frames, and is referred to as the *constant QP* (CQP) algorithm in this thesis.

The second reference algorithm is referred to as the *optimum QP* (OQP) algorithm in this thesis. The QP is obtained by first quantizing/dequantizing/IDCT the frame 31 times, and then selecting the QP that produces the closest PSNR to the target PSNR. The term *optimum* is used because it is truly the best quality achievable for a frame-level quality control algorithm (one QP per frame).

### 5.2.4. Simulation Results and Analysis

Two CIF (352x240) test sequences (*table tennis* and *football*) of 151 frames are used in the simulations. Three types of GOV structures are used: Pure I-VOP, IPPP... (only the first frame is I-VOP), and IBBPBBP... (with intra period of 15). For each GOV structure, these sequences are first MPEG-4 coded using the CQP algorithm with QP = 5,

10, 15, 20, 25, and 30. The average luminance-channel PSNR of the encoded sequence is used as the target PSNR for the other three algorithms. Simulation results of each GOV structure are shown in Table 5.1 to Table 5.3. Four sample PSNR/QP graphs are shown in Figure 5.1 to Figure 5.4. Note that the shaded columns in all tables indicate the target PSNR used by the other three algorithms. The bold typeface in the tables indicates situations when the average PSNR is the same as the target PSNR (with 2 decimal precision) and when the PSNR variance is zero (with 2 decimal precision).

Table 5.1: Pure I-VOP PSNR (dB) Mean and Variance over the Entire Sequence of CQP, OQP, FLCQ VBR and MVCQ VBR algorithms

| QP | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|
| | CQP | OQP | FLCQ VBR | MVCQ VBR | CQP | OQP | FLCQ VBR | MVCQ VBR |
| Table Tennis | | | | | | | | |
| 5 | 36.59 | 36.57 | 36.60 | **36.59** | 0.16 | 0.16 | 0.18 | **0.00** |
| 10 | 32.13 | 32.03 | 32.04 | **32.13** | 1.24 | 0.03 | 0.04 | **0.00** |
| 15 | 29.99 | 29.97 | 30.06 | **29.99** | 1.60 | 0.01 | 0.01 | **0.00** |
| 20 | 28.63 | 28.66 | 28.70 | 28.64 | 1.40 | 0.01 | 0.01 | **0.00** |
| 25 | 27.59 | 27.74 | 27.76 | 27.75 | 1.18 | 0.15 | 0.15 | 0.13 |
| 30 | 26.75 | 27.12 | 27.15 | 27.12 | 1.11 | 0.47 | 0.45 | 0.44 |
| Football | | | | | | | | |
| 5 | 36.81 | 36.45 | 36.52 | **36.81** | 0.51 | 0.09 | 0.18 | **0.00** |
| 10 | 32.14 | 31.99 | 31.99 | **32.14** | 1.55 | 0.04 | 0.05 | **0.00** |
| 15 | 29.72 | 29.77 | 29.74 | 29.74 | 1.96 | 0.05 | 0.06 | 0.02 |
| 20 | 28.22 | 28.33 | 28.34 | 28.32 | 2.11 | 0.19 | 0.19 | 0.18 |
| 25 | 27.10 | 27.35 | 27.36 | 27.34 | 2.19 | 0.47 | 0.47 | 0.46 |
| 30 | 26.21 | 26.77 | 26.78 | 26.77 | 2.18 | 0.93 | 0.93 | 0.91 |

Table 5.2: IPPP PSNR (dB) Mean and Variance over the Entire Sequence of CQP, OQP, FLCQ VBR and MVCQ VBR algorithms

| QP | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|
| | CQP | OQP | FLCQ VBR | MVCQ VBR | CQP | OQP | FLCQ VBR | MVCQ VBR |
| Table Tennis | | | | | | | | |
| 5 | 35.62 | 35.75 | 35.80 | **35.62** | 0.39 | 0.09 | 0.09 | **0.00** |
| 10 | 31.37 | 31.40 | 31.40 | **31.37** | 1.24 | 0.02 | 0.04 | **0.00** |
| 15 | 29.49 | 29.51 | 29.61 | 29.50 | 1.41 | 0.01 | 0.13 | **0.00** |
| 20 | 28.21 | 28.35 | 28.74 | 28.30 | 1.19 | 0.06 | 0.37 | 0.04 |
| 25 | 27.35 | 27.73 | 28.23 | 27.60 | 1.07 | 0.20 | 0.53 | 0.21 |
| 30 | 26.59 | 27.28 | 27.97 | 27.07 | 1.14 | 0.44 | 0.96 | 0.54 |
| Football | | | | | | | | |
| 5 | 35.48 | 35.55 | 35.74 | 35.49 | 0.64 | 0.25 | 0.25 | **0.00** |
| 10 | 30.85 | 30.87 | 30.84 | 30.86 | 1.28 | 0.02 | 0.02 | **0.00** |
| 15 | 28.77 | 28.85 | 28.82 | 28.80 | 1.54 | 0.04 | 0.05 | 0.04 |
| 20 | 27.37 | 27.49 | 27.48 | 27.47 | 1.56 | 0.18 | 0.18 | 0.19 |
| 25 | 26.44 | 26.67 | 26.67 | 26.66 | 1.54 | 0.39 | 0.40 | 0.40 |
| 30 | 25.69 | 26.18 | 26.18 | 26.18 | 1.54 | 0.72 | 0.72 | 0.72 |

Table 5.3: IBBP PSNR (dB) Mean and Variance over the Entire Sequence of CQP, OQP, FLCQ and MVCQ algorithms

| QP | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|
| | CQP | OQP | FLCQ VBR | MVCQ VBR | CQP | OQP | FLCQ VBR | MVCQ VBR |
| Table Tennis | | | | | | | | |
| 5 | 35.72 | 35.77 | 35.73 | **35.72** | 0.45 | 0.06 | 0.09 | **0.00** |
| 10 | 31.53 | 31.52 | 31.58 | **31.53** | 1.30 | 0.01 | 0.08 | **0.00** |
| 15 | 29.63 | 29.62 | 29.73 | 29.65 | 1.54 | 0.01 | 0.09 | **0.00** |
| 20 | 28.38 | 28.48 | 28.63 | 28.50 | 1.29 | 0.06 | 0.13 | 0.06 |
| 25 | 27.53 | 27.79 | 27.94 | 27.81 | 1.20 | 0.25 | 0.28 | 0.24 |
| 30 | 26.82 | 27.27 | 27.42 | 27.30 | 1.20 | 0.55 | 0.54 | 0.54 |
| Football | | | | | | | | |
| 5 | 35.59 | 35.77 | 36.00 | **35.59** | 0.73 | 0.19 | 0.13 | **0.00** |
| 10 | 30.94 | 30.95 | 31.00 | **30.94** | 1.35 | 0.02 | 0.03 | **0.00** |
| 15 | 28.83 | 28.83 | 28.86 | 28.86 | 1.49 | 0.06 | 0.07 | 0.06 |
| 20 | 27.42 | 27.52 | 27.52 | 27.51 | 1.47 | 0.17 | 0.19 | 0.17 |
| 25 | 26.49 | 26.69 | 26.70 | 26.70 | 1.44 | 0.38 | 0.39 | 0.38 |
| 30 | 25.73 | 26.17 | 26.18 | 26.18 | 1.43 | 0.70 | 0.70 | 0.70 |

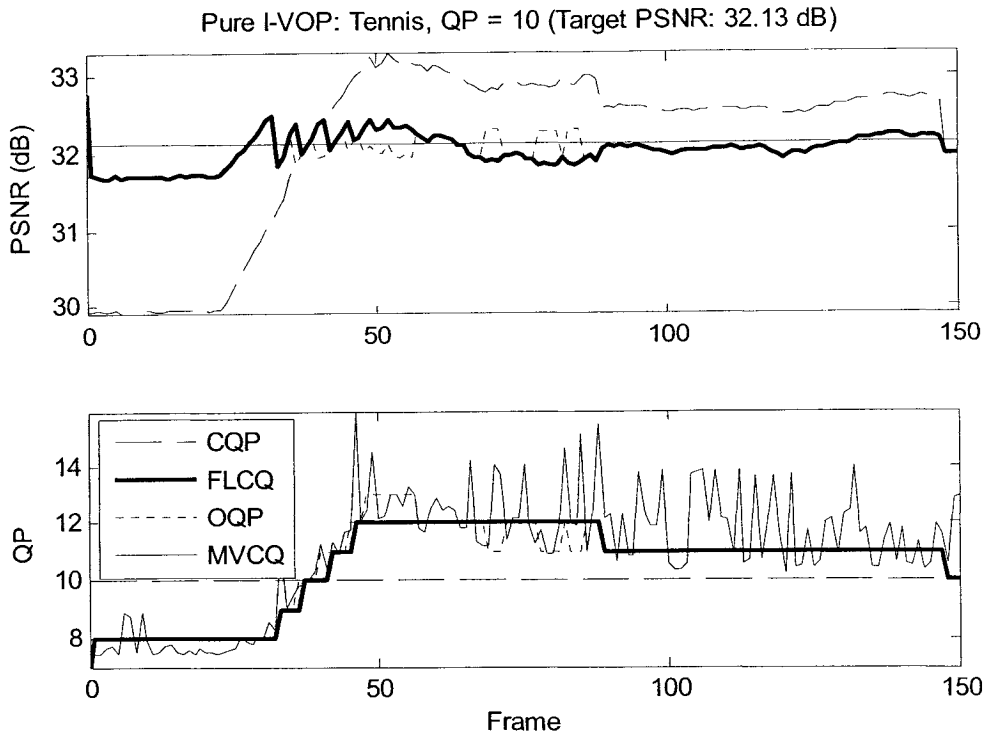



Figure 5.1: PSNR and QP of CQP, OQP, FLCQ VBR, and MVCQ VBR algorithms for Tennis (GOV: Pure I-VOP, Target Quality: 32.13 dB)

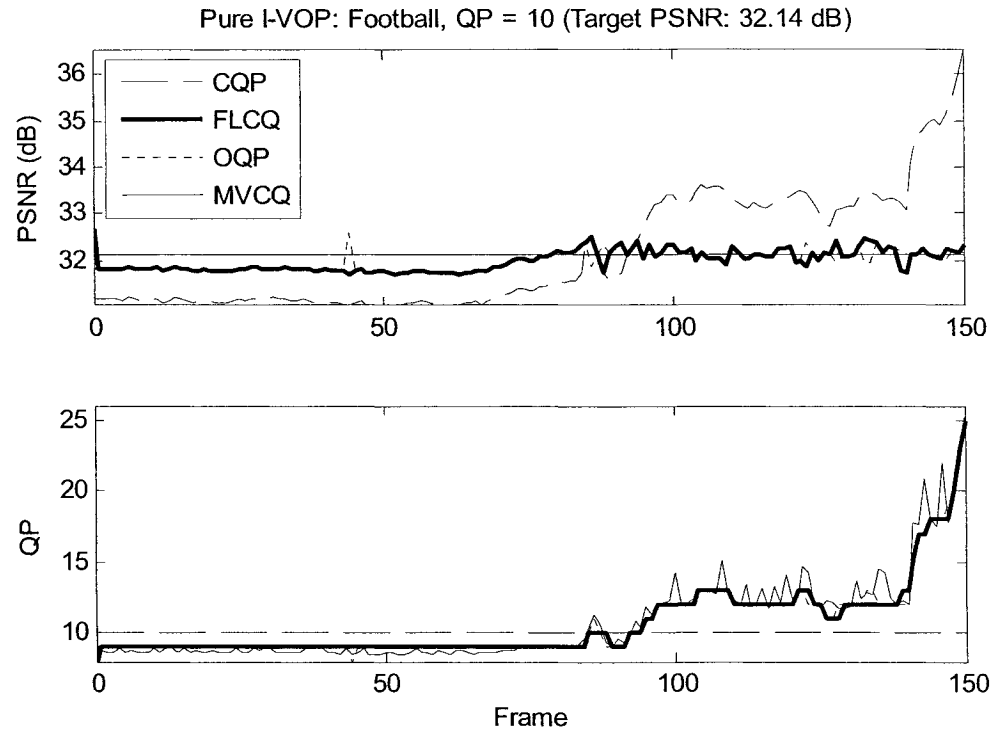



Figure 5.2: PSNR and QP of CQP, OQP, FLCQ VBR, and MVCQ VBR algorithms for Football (GOV: Pure I-VOP, Target Quality: 32.14 dB)

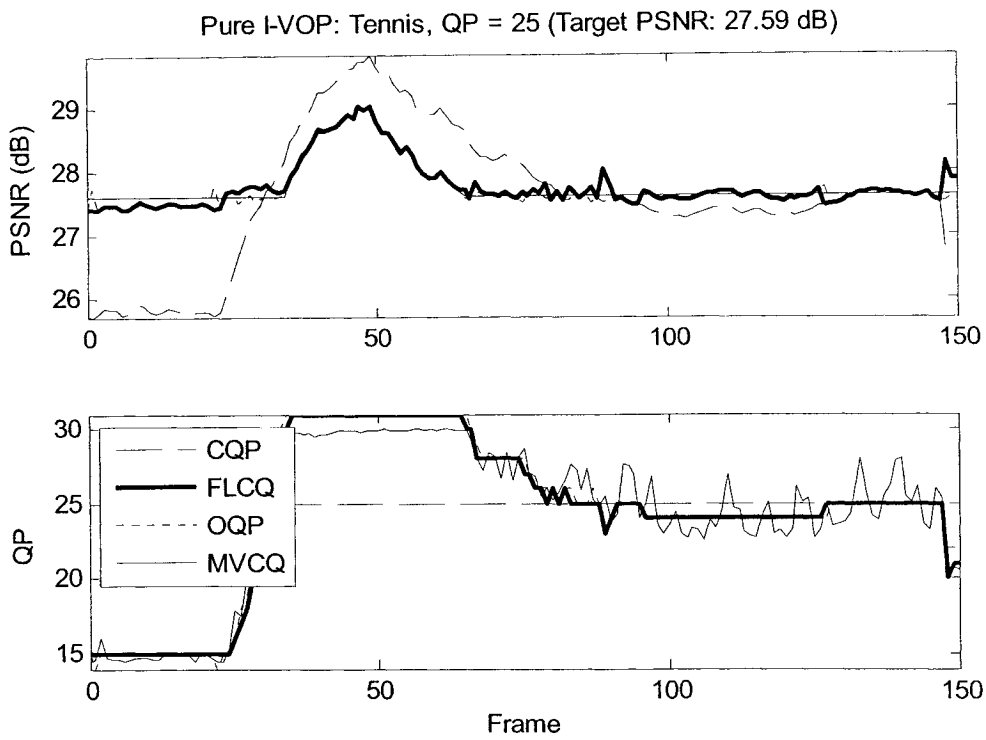



Figure 5.3: PSNR and QP of CQP, OQP, FLCQ VBR, and MVCQ VBR algorithms for Tennis (GOV: Pure I-VOP, Target Quality: 27.59 dB)

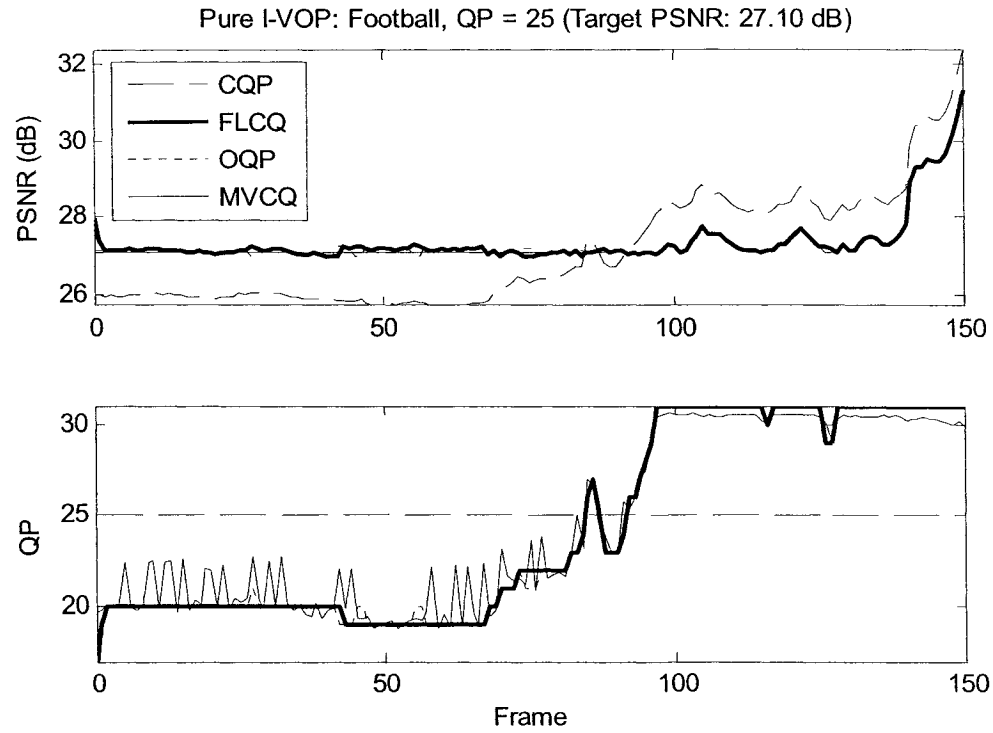



Figure 5.4: PSNR and QP of CQP, OQP, FLCQ VBR, and MVCQ VBR algorithms for Football (GOV: Pure I-VOP, Target Quality: 27.10 dB)

As shown in Table 5.1 to Table 5.3, the proposed FLCQ VBR algorithm outperforms the CQP reference algorithm significantly as the quality variance is reduced significantly for all GOV structures.

When the FLCQ VBR algorithm is compared against the OQP algorithm, both algorithms control the quality at the frame level. The FLCQ VBR algorithm uses the Laplacian distortion model to estimate the distortion, which requires very little computation. On the other hand, the OQP method needs to quantize and de-quantize the frame 31 times to obtain the optimum QP, which is much more complex and time consuming. As expected, the performance of the FLCQ VBR algorithm is slightly worse than the OQP algorithm. However, its computational complexity is significantly lower than OQP.

For the MVCQ VBR algorithm, as shown in the tables, the performance is quite good. It meets the target PSNR with almost no error, and zero variance (in the order of $10^{-10}$) is achieved in many cases as highlighted in bold typeface in the tables and in Figure 5.1 and Figure 5.2. In fact, the non-zero variance is caused by running outside of the quantization parameter limits (see Figure 5.3 and Figure 5.4). Since one cannot set the QP larger than 31 (for 5-bit precision quantizer), it is not possible to lower the PSNR for some macroblocks/frames to the target, hence the variance increases. Note that this phenomenon does not happen for small QPs, because it is possible to increase the QPs in order to lower the PSNR. As QP increases (target PSNR decreases), there is less freedom to change the QP. Eventually, when QP reaches 31, it is not possible to lower the PSNR any further to the target PSNR.

As a final observation, because the MVCQ VBR algorithm relies on the OQP method to set up the initial target MB distortions, its performance is always better than the OQP method for pure I-VOP cases. For IPPP and IBBP cases, because of the dependencies among intra and inter frames, it is more difficult to compare the results. However, in general, the MVCQ VBR algorithm gives better/similar performance compared to the OQP method with little extra computation (running the Viterbi algorithm).

## 5.3. MPEG-4 Constant-Quality CBR Algorithms with Fixed Target Quality

The CQ VBR algorithms presented in the previous section achieve true constant quality by ignoring the buffer constraint. That assumption is valid for some offline or high bandwidth applications, but is not suitable for applications with limited buffer size. For such applications, it is critical that buffer overflow should never occur. Buffer overflow would lead to loss of information and thus a corrupted bitstream. Therefore, the algorithms proposed in this section are designed to avoid buffer overflow. Buffer underflow is *not* prevented. These algorithms aim at producing CBR bitstreams that meet a fixed target quality (FTQ) whenever possible. The target quality is also the highest quality that these algorithms will produce.

Two CQ CBR algorithms are proposed in this section. Similar to the previous section, the FLCQ CBR algorithm with FTQ [99] controls the quality at the frame level, and is presented in Section 5.3.1. The MVCQ CBR algorithm with FTQ [99] controls the

quality at the macroblock level, and is presented in Section 0. Besides using different QM/BM algorithms, these two algorithms share the same overall control structure to achieve CQ while avoiding buffer overflow for CBR channels. Simulation results and discussions are presented in Section 5.3.3.

### 5.3.1. MPEG-4 Frame-level Laplacian Constant-Quality (FLCQ) CBR Algorithm with Fixed Target Quality (FTQ)

The first CQ-CBR algorithm is called Frame-level Laplacian Constant Quality FTQ (FLCQ FTQ) CBR algorithm. It controls the quality at the frame level (one QP per frame) by using the FLQM and FLBM algorithms developed in Section 5.1.1 and 5.1.2. The FLCQ FTQ algorithm is described as follows.

**Step 1**: Compute the target PSNR ($PSNR_T$) for the current frame:

a) For the first frame, set $PSNR_T = PSNR_{Target}$, where $PSNR_{Target}$ is the constant quality goal for the entire sequence. Go to **Step 2**.

b) Since the purpose of the algorithm is not only to keep the quality constant, but also to avoid buffer overflow, $PSNR_T$ needs to be adjusted based on the buffer occupancy ($0 \le Buffer_{Ratio} \le 1$). With this in mind,

$$PSNR_T = \min\left(\kappa \times PSNR_{SlidingWindow}, PSNR_{Target}\right) \tag{49}$$

where $PSNR_{SlidingWindow}$ is the average PSNR of the $W$ frames coded in the past, and $\kappa$ is a scaling factor affected by the buffer occupancy, and is defined as

$$\kappa = \min\left(\max\left(\kappa_1, \kappa_{\min}\right), \kappa_{\max}\right)$$
$$\kappa_1 = \begin{cases} \kappa_{\max} & Buffer_{Ratio} = 0 \\ Buffer_{Target} / Buffer_{Ratio} & 0 < Buffer_{Ratio} \le 1 \end{cases} \quad (50)$$

where $\kappa_{min}$ ($\kappa_{min}<1$) and $\kappa_{max}$ ($\kappa_{max}>1$) are the minimum and maximum scaling factor respectively, and $Buffer_{Target}$ is the target buffer level. In simulations, $W$, $\kappa_{min}$, $\kappa_{max}$, and $Buffer_{Target}$ are set to be 5, 0.8, 1.2, and 0.8 respectively.

**Step 2**: Use the FLQM algorithm defined in Section 5.1.1 to set the frame QP with $MSE_{Target}$ set to be the MSE corresponding to $PSNR_T$, i.e.,

$MSE_{Target} = 255^2 / 10^{PSNR_T / 10}$.

**Step 3**: Use the Laplacian rate model developed in Section 3.1.3 to estimate the bits for the QP selected in Step 2.

**Step 4**: If the buffer is expected to exceed the critical ratio ($b_{Critical}$=95% in the simulations) after adding this frame (with the QP set in Step 2) and removing the average bit per frame, i.e.,

$$BufferOccupancy + EstimatedBit - \frac{TargetBitrate}{FramePerSecond} > b_{Critical} \times BufferSize \quad (51)$$

Then, use the FLBM to set the new QP(s) for the bit budget ($Bit_{Target}$) given by

$$Bit_{Target} = \max\left\{\begin{matrix} BitPerFrame, \\ b_{Critical} \times BufferSize - BufferOccupancy + BitPerFrame \end{matrix}\right\} \quad (52)$$

### 5.3.2. MPEG-4 Macroblock-level Viterbi Constant-Quality (MVCQ) CBR Algorithm with Fixed Target Quality (FTQ)

The second CQ-CBR algorithm is referred to as Macroblock-level Viterbi Constant Quality FTQ (MVCQ FTQ) CBR algorithm. It controls the quality in the macroblock level (one QP per macroblock) by using the MVQM and MVBM algorithms derived in Section 5.1.3 and 5.1.4.

The MVCQ FTQ algorithm shares the same control structure as the FLCQ FTQ algorithm. It is obtained by simply changing the QM/BM algorithms used in the FLCQ FTQ algorithm, i.e., replacing FLQM by MVQM (in Step 2) and FLBM by MVBM (in Step 4). Note that in Step 3, instead of using frame QP, the average macroblock QP is used in the Laplacian rate model to estimate the bits.

### 5.3.3. Simulation Results and Analysis

To evaluate the performance of the proposed CBR algorithms, Q2 [7] (frame-level control) and TM5 (Test Model 5) [8] (macroblock-level control) are used as the reference algorithms. Four test sequences (table tennis, football, flower garden, and mobile calendar) of 150-frame long and 30 frame/sec are used in the simulations. The first three sequences have frame size 352x240, and the last sequence has frame size 352x288. Each sequence is MPEG-4 encoded using three GOV structures (pure I-VOP, IPPP with intra period of 30 frames, and IBBP with intra period of 15 frames) at 1000, 2000, 3000, and 4000 kbps. The buffer sizes are set to be 0.5 second (15 frames) of the target bitrate. For each sequence and bitrate, five target PSNRs are tested ($PSNR_{Target}$ =

max($PSNR_{Q2}$, $PSNR_{TM5}$), $PSNR_{Target}$±0.5dB, and $PSNR_{Target}$±1.0dB) for FLCQ FTQ and MVCQ FTQ algorithms. However, due to the large amount of data, Table 5.4 and Table 5.5 only show the simulation results for IPPP and IBBP GOV structures at 1000, 2000, and 4000 kbps with target PSNR (shaded) set to be the highest average PSNR of Q2 and TM5. Two sample plots are also shown in Figure 5.5 and Figure 5.6 that provide information on PSNR, bits, and buffer occupancy for each frame over time. The vertical dotted lines in these figures are GOV boundaries (location of the I-VOP).

To make a fair comparison, the frame-level QP control algorithms (Q2 and FLCQ FTQ) and macroblock-level QP control algorithms (TM5 and MVCQ FTQ) are compared separately. In most instances, as demonstrated in the results, within each category, the proposed algorithms provide a similar average PSNR while having lower PSNR variance and lower bitrate than the reference algorithms. For some instances, (close to) zero PSNR variance is achieved. Note that due to the constraint defined in (49), the target PSNR for each frame is always smaller than or equal to the $PSNR_{Target}$ (shaded). Hence, the average PSNR of the proposed algorithm is always smaller than or equal to $PSNR_{Target}$. When combined with (51), another consequence of (49) is that the proposed algorithms tend to have lower actual bitrate than the target bitrate. This is because the algorithms will not spend more bits if the target PSNR can be achieved (even if the buffer occupancy is low), but will spend fewer bits if buffer occupancy is high. In other words, the proposed algorithms will prevent buffer overflow, but not buffer underflow.

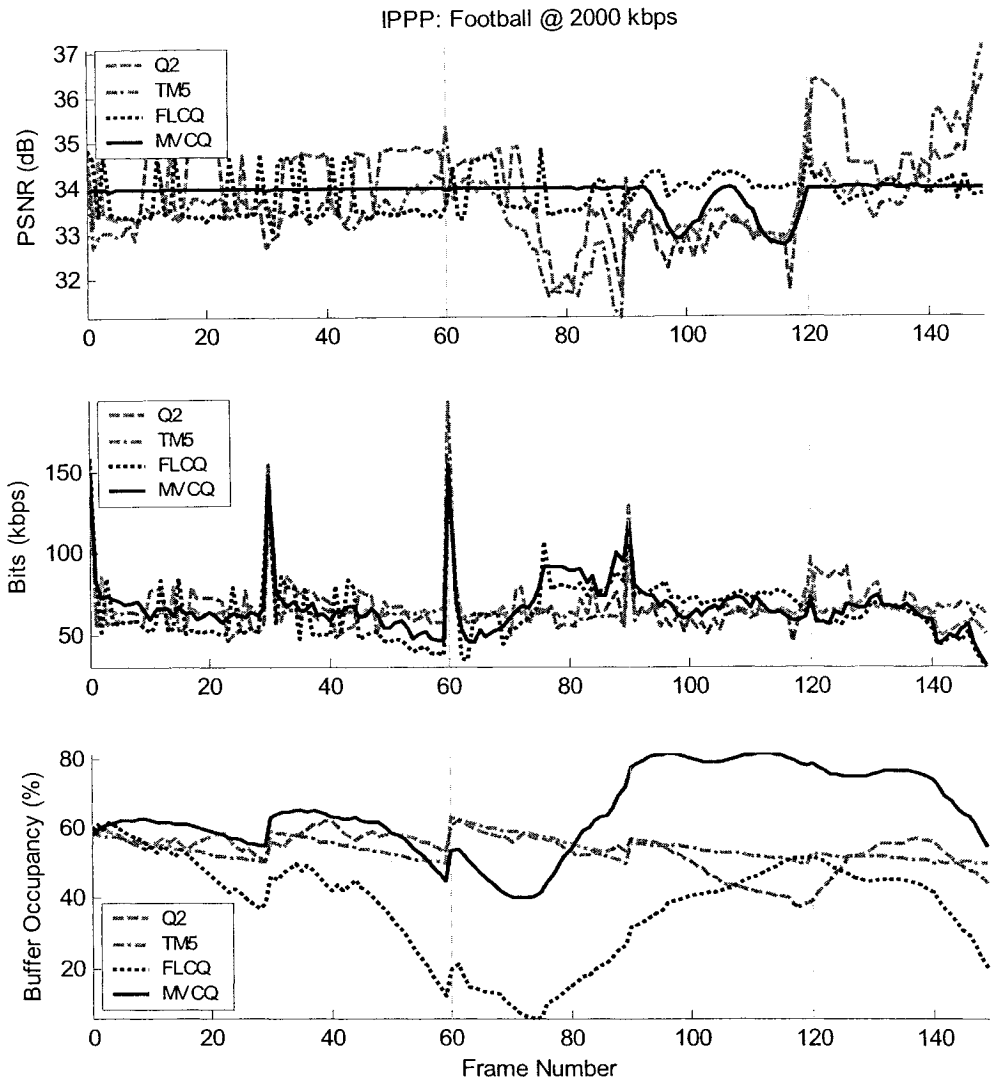



Figure 5.5: PSNR, bits per frame, and buffer occupancy plots of football for Q2, TM5, FLCQ FTQ, and MVCQ FTQ algorithms (GOV: IPPP, Rate: 2000 kbps) [99]

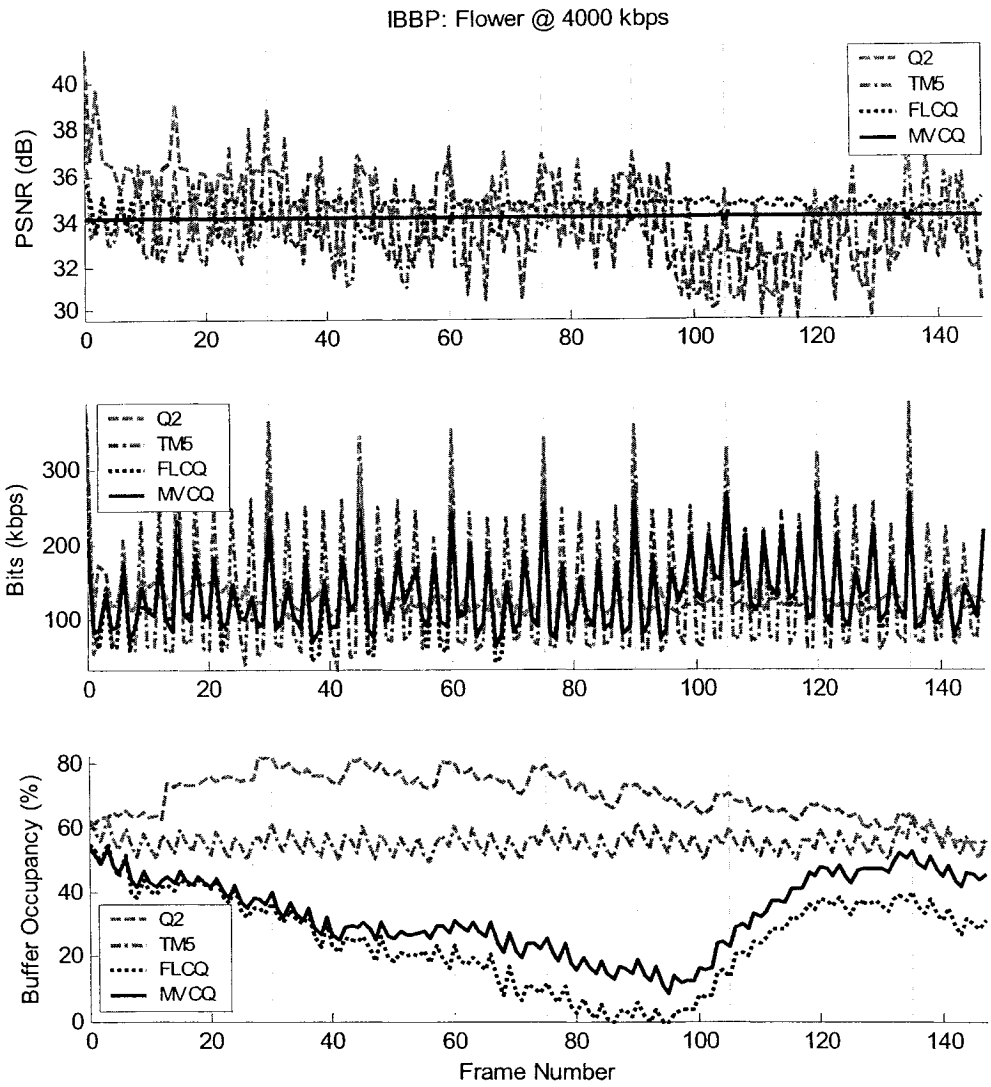



Figure 5.6: PSNR, bits/frame, and buffer occupancy plots of flower garden for Q2, TM5, FLCQ FTQ, and MVCQ FTQ algorithms (GOV: IBBP, Rate: 4000 kbps) [99]

Table 5.4: Selected Simulation Results for IPPP GOV Structure for sequences table tennis (S1), football (S2), flower garden (S3), and mobile calendar (S4) [99]

| | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 |
| Target Bitrate: 1000 kbps | | | | | | | | | | | | |
| Q2 | 1187 | 996 | 1001 | 1002 | 34.65 | 30.53 | 25.84 | 26.75 | 5.93 | 0.86 | 0.96 | 0.25 |
| TM5 | 1006 | 999 | 1001 | 1000 | 33.30 | 30.36 | 25.14 | 26.31 | 5.83 | 0.80 | 0.87 | 0.15 |
| FLCQ FTQ | 880 | 1000 | 1005 | 995 | 33.44 | 30.57 | 25.86 | 26.78 | 2.28 | 0.05 | 0.03 | 0.02 |
| MVCQ FTQ | 878 | 1001 | 1021 | 1014 | 33.33 | 30.48 | 25.72 | 26.69 | 2.68 | 0.04 | 0.24 | 0.10 |
| Target Bitrate: 2000 kbps | | | | | | | | | | | | |
| Q2 | 1859 | 1989 | 2003 | 1999 | 37.03 | 33.95 | 29.64 | 30.27 | 3.52 | 1.07 | 1.12 | 0.20 |
| TM5 | 2001 | 1999 | 2000 | 2000 | 36.63 | 33.51 | 28.74 | 29.58 | 4.92 | 0.76 | 1.24 | 0.17 |
| FLCQ FTQ | 1663 | 1941 | 1990 | 2015 | 36.50 | 33.84 | 29.61 | 30.35 | 1.71 | 0.21 | 0.07 | 0.03 |
| MVCQ FTQ | 1654 | 2009 | 2057 | 2041 | 36.31 | 33.85 | 29.56 | 30.20 | 1.69 | 0.09 | 0.14 | 0.15 |
| Target Bitrate: 4000 kbps | | | | | | | | | | | | |
| Q2 | 2802 | 3983 | 4001 | 4017 | 39.49 | 37.81 | 34.62 | 34.84 | 0.34 | 2.16 | 1.96 | 0.73 |
| TM5 | 3996 | 3999 | 4001 | 4000 | 39.94 | 37.04 | 33.39 | 33.65 | 2.31 | 0.73 | 1.55 | 0.26 |
| FLCQ FTQ | 2799 | 3963 | 3960 | 4097 | 39.50 | 37.73 | 34.68 | 35.02 | 0.14 | 1.27 | 0.05 | 0.02 |
| MVCQ FTQ | 3170 | 4055 | 4127 | 4121 | 39.79 | 37.46 | 34.46 | 34.62 | 0.43 | 0.30 | 0.36 | 0.21 |

Table 5.5: Selected Simulation Results for IBBP GOV Structure for sequences table tennis (S1), football (S2), flower garden (S3), and mobile calendar (S4) [99]

| | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 | S1 | S2 | S3 | S4 |
| Target Bitrate: 1000 kbps | | | | | | | | | | | | |
| Q2 | 1002 | 1002 | 1011 | 1007 | 32.55 | 29.80 | 24.82 | 26.24 | 1.54 | 1.32 | 1.94 | 1.19 |
| TM5 | 1000 | 1000 | 1001 | 1000 | 33.76 | 30.25 | 26.11 | 27.88 | 4.26 | 2.52 | 0.91 | 0.50 |
| FLCQ FTQ | 839 | 994 | 999 | 1021 | 33.32 | 30.17 | 26.11 | 27.66 | 1.20 | 0.13 | 0.02 | 0.27 |
| MVCQ FTQ | 865 | 1000 | 1012 | 1027 | 33.18 | 30.07 | 26.09 | 27.62 | 1.03 | 0.23 | 0.02 | 0.30 |
| Target Bitrate: 2000 kbps | | | | | | | | | | | | |
| Q2 | 1928 | 2004 | 2014 | 2015 | 36.75 | 33.53 | 29.02 | 30.52 | 3.75 | 2.13 | 3.04 | 1.09 |
| TM5 | 2001 | 2000 | 2001 | 2001 | 36.84 | 33.33 | 29.61 | 30.93 | 4.55 | 3.29 | 1.90 | 0.98 |
| FLCQ FTQ | 1711 | 1968 | 1933 | 1985 | 36.42 | 33.45 | 29.62 | 30.95 | 1.03 | 0.14 | 0.06 | 0.04 |
| MVCQ FTQ | 1707 | 1995 | 2018 | 2010 | 36.19 | 33.37 | 29.62 | 30.94 | 1.32 | 0.20 | 0.00 | 0.00 |
| Target Bitrate: 4000 kbps | | | | | | | | | | | | |
| Q2 | 2903 | 4021 | 4021 | 4010 | 39.39 | 37.44 | 34.15 | 35.33 | 0.21 | 3.15 | 4.25 | 2.74 |
| TM5 | 4000 | 4000 | 4001 | 4001 | 39.93 | 36.76 | 33.69 | 34.59 | 4.06 | 4.24 | 3.58 | 2.17 |
| FLCQ FTQ | 3000 | 3775 | 3893 | 3729 | 39.80 | 37.30 | 34.56 | 35.13 | 0.42 | 1.02 | 0.26 | 0.05 |
| MVCQ FTQ | 3338 | 4036 | 3975 | 4073 | 39.60 | 37.18 | 34.16 | 35.34 | 0.69 | 0.28 | 0.00 | 0.00 |

Note that when the buffer permits, the MVCQ FTQ algorithm meets the PSNR more accurately and gives lower PSNR variance than the FLCQ FTQ algorithm since it can change the QPs at the macroblock level. Thus, (close to) zero PSNR variance is often produced by the MVCQ FTQ algorithm, but rarely by the FLCQ FTQ algorithm (see Table 5.5 and Figure 5.6). On the other hand, because of the extra degree of freedom in

macroblock QP control, extra headers are inserted to keep track the QP changes. Thus, in general, macroblock QP control algorithms spend bits less efficiently than frame-level QP controls. As a result, for the same bitrate, frame-level QP control algorithms tend to give higher average PSNR than the macroblock QP control algorithms. An example of this phenomenon can be observed in the buffer occupancy plot in the bottom of Figure 5.5. In that figure, the buffer occupancy of the MVCQ FTQ algorithm is always higher than the FLCQ FTQ algorithm. One undesirable implication of this is that the MVCQ FTQ algorithm is more likely to reach a fuller buffer state. This may force a reduction in PSNR due to (50). The PSNR curve in Figure 5.5 between frame 90 and 120 is an example of this effect.

Since the actual PSNRs given by MVCQ FTQ and FLCQ FTQ algorithms are generally smaller than the target PSNR, one easy way to raise the average PSNR is to raise the target PSNR. Although not shown in Table 5.4, simulation results suggest that by increasing the target PSNR, the average PSNR can also be increased (up to a certain point). However, the PSNR variance would also be increased at the same time. A higher target PSNR usually requires the spending of more bits and causing the buffer to reach high status faster (with the same bitrate and buffer size). When the buffer is high, the target frame PSNR is forced to drop, and this causes an increase in PSNR variance. Thus, an increase in target PSNR is usually followed by an increase in both average quality and quality variation.

As a final remark, setting good target PSNRs is not trivial. Decreasing the target PSNR may lower the PSNR variance, but it may also lead to a lower video quality and waste of channel bandwidth. On the other hand, increasing the target PSNR usually

provides better average quality with the expense of higher quality variance. For both CQ algorithms proposed in this section, the target PSNR is a constant value (maximum average PSNR of Q2 and TM5) determined before encoding. However, whether it is too low or too high is unknown until the sequence is actually encoded. Thus, the two CQ CBR algorithms presented in the next section are designed to solve this issue.

## 5.4. MPEG-4 Constant-Quality CBR Control Algorithms with Dynamic Target Quality

In the previous section, the CQ CBR algorithms use a fixed target quality for the entire sequence. As mentioned in Section 5.3.3, setting a *good* target quality for the entire sequence is not an easy job. Because the encoding difficulty changes for different sequences, it is unpractical to find one single *good* target quality for all sequences even if they were encoded at the same bitrate. Also, since the encoding difficulty changes for different parts of one sequence, a *good* target quality for one part of a sequence may not be as *good* for another part of the same sequence. To handle this issue, the target quality is adjusted dynamically as outlined in this section. Note that one undesirable effect of changing target quality dynamically is that the quality variance would also increase.

Both CQ CBR algorithms presented in this section use a dynamic target quality (DTQ) [115]. They are developed from the FTQ algorithms proposed in the previous section. The only difference is in Step 1, where the target quality is computed.

Similar to the previous section, the FLCQ CBR algorithm with DTQ [115] controls the quality at the frame level (one QP per frame). The MVCQ CBR algorithm

with DTQ [115] controls the quality at the macroblock level (one QP per macroblock). Besides using different QM/BM algorithms, these two algorithms share the same overall control structure to achieve CQ while preventing buffer overflow. The FLCQ/MVCQ DTQ algorithms are described as follows.

**Step 1**: Compute the target PSNR for the current frame (see Section 5.4.1 below for details).

**Step 2**: Use the target PSNR set in Step 1 and the QM algorithm (FLQM for FLCQ DTQ, and MVQM for MVCQ DTQ) to set the frame/macroblock QP(s).

**Step 3**: Use the Laplacian rate model (see Section 3.1.3) to estimate the bits for the QP(s) computed in Step 2. Note for MVCQ DTQ, the average QP is used by the rate model.

**Step 4**: If the buffer is not expected to exceed the critical ratio ($b_{Critical}$=95% in simulations) after adding this frame (with the QP(s) set in Step 2 and the bits estimated in Step 3) and removing the average bits per frame, i.e.,

$$BufferOccupancy + EstimatedBit - BitPerFrame \le b_{Critical} \times BufferSize$$

$$BitPerFrame = \frac{TargetBitrate}{FramePerSecond}$$

Then, end the algorithm with output QPs as computed in Step 2.

Otherwise (buffer exceeding critical ratio), use the BM algorithm (FLBM for FLCQ DTQ, and MVBM for MVCQ DTQ) to set the new QP(s) for the bit budget ($Bit_{Target}$) given by

$$Bit_{Target} = \max\left\{\begin{array}{l} BitPerFrame, \\ b_{Critical} \times BufferSize - BufferOccupancy + BitPerFrame \end{array}\right\} \quad (53)$$

Then, end the algorithm with the new QPs.

### 5.4.1. Setting the Target PSNR

As mentioned in the introduction of this chapter, a CQ CBR control algorithm should minimize the PSNR variance while meeting the target bitrate and maximizing the PSNR. In many cases, not all goals can be satisfied simultaneously. Thus, a weighting factor for these conflicting goals is used. The cost function ($\tilde{C}_k$) for frame $k$ is defined as

$$\tilde{C}_k = -\lambda_1 S(b) T_k^2 + \left\{ \frac{1}{W-1}\left(\sum_{i=0}^{W-1} P_{k-i}^2\right) - \frac{1}{W(W-1)}\left(\sum_{i=0}^{W-1} P_{k-i}\right)^2 \right\}, \lambda_1 \geq 0, W > 0 \qquad (54)$$

where $T_k$ and $P_k$ are the target and actual PSNR of frame $k$ respectively, $S(b)$ is a function of buffer occupancy ($b$), $W$ is the sliding window size used to estimate the sample variance, and $\lambda_1$ is the weighting factor.

The first term of (54) is used to maximize the target PSNR ($-T_k^2$) while maximizing the buffer utilization $S(b)$ defined as

$$S(b) = \begin{cases} S_{Low} \dfrac{b_{Low} - b}{b_{Low}} & 0 \leq b < b_{Low} \\ 0 & b_{Low} \leq b \leq b_{High} \\ -S_{High} \dfrac{b - b_{High}}{b_{Critical} - b_{High}} & b_{High} \leq b < b_{Critical} \end{cases} \qquad (55)$$

where $b_{Low}/b_{High}/b_{Critical}$ are thresholds when the buffer is low/high/very high, and $S_{Low}/S_{High}$ are the scaling factor when the buffer is low/high. A sample plot of $S(b)$ function is shown in Figure 5.7. Note that when the buffer is low, $S(b)$ is positive, and the target PSNR tends to increase. On the other hand, when the buffer is high, $S(b)$ is negative, and the target PSNR tends to decrease. Thus, (55) means that the $T_k$ should be increased/not changed/decreased if buffer is low/normal/high. In the simulations, $b_{Low}$=0.2, $b_{High}$=0.7, $b_{Critical}$=0.9, $S_{Low}$=$S_{High}$=0.5.

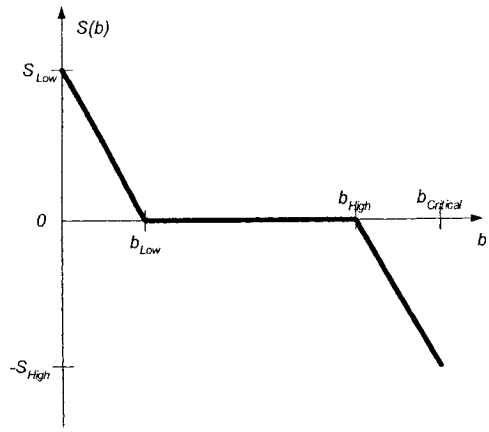



Figure 5.7: Buffer utilization function $S(b)$

The second term of (54) is the sample variance [116] of the actual PSNRs within the sliding window. Note that instead of computing the sample variance from the beginning of the sequence, only the current and the past $W$-1 frames inside the sliding window are used. A smaller window size would be appropriate if the sequence is changing rapidly, but would result in a less accurate estimation of the variance of a slowly changing sequence. In the simulations, $W$=30 is used.

Since the actual PSNR ($P_k$) of the current frame (frame $k$) cannot be known before actually encoding the frame, it is assumed that $P_k \cong T_k$. As a result, (54) can be approximated as

$$C_k = -\lambda_1 S(b) T_k^2 + \left\{ \frac{1}{W-1}\left( T_k^2 + \sum_{i=1}^{W-1} P_{k-i}^2 \right) - \frac{1}{W(W-1)}\left( T_k + \sum_{i=1}^{W-1} P_{k-i} \right)^2 \right\} \tag{56}$$

Minimizing $C_k$ by setting $\frac{\partial C_k}{\partial T_k} = 0$ and $\frac{\partial^2 C_k}{\partial T_k^2} > 0$, the best target quality $T_k$ for frame $k$ is

$$T_k = \frac{1}{1-\lambda_1 WS(b)}\left( \frac{1}{W-1} \sum_{i=1}^{W-1} P_{k-i} \right) \text{ and } 0 \le \lambda_1 < \frac{1}{WS(b)} \tag{57}$$

Because $\lambda_1$ is a function of $S(b)$ and $W$, it is non-trivial to determine the range of $\lambda_1$ in practice. Thus, in the simulations, instead of using $\lambda_1$ directly, a normalized $\lambda$ ($\lambda = \lambda_1 / \lambda_{max}$) is used, where $\lambda_{max} = \frac{1}{W \cdot \max\{S(b)\}}$ and $0 \le \lambda < 1$.

Note that when $\lambda$=0, only the second term remains in (56), and the goal is to minimize the variance disregarding the buffer usage (except for buffer overflow that is handled in Step 4). When $\lambda \approx 1$, the first term will dominate (56). Hence, the goal becomes maximizing PSNR and buffer utilization, which is like the traditional CBR algorithm. Thus, as $\lambda$ increases from 0 to (nearly) 1, the aim of the cost function changes from purely constant-quality (and not caring too much about the rate) to pure constant-bit-rate (and not caring too much about the variation in quality). In other words, $\lambda$ provides an extra degree of freedom to control quality variations that is not present in the traditional CBR and CQ algorithms.

### 5.4.2. Computational Complexity of FLCQ DTQ and MVCQ DTQ

Since there are no loops in FLCQ DTQ and MVCQ DTQ, the computational complexity of these algorithms is the complexity of FLQM/MVQM plus the complexity of FLBM/MVBM. In FLQM, FLBM, and MVQM, the Laplacian rate/distortion models are used. Since the Laplacian models use very few computations, the computational complexities of FLQM, FLBM, and MVQM are negligible. For the MVCQ DTQ algorithm, due to the complexity in the MVQM algorithm, it has a higher computational complexity. Note the computational complexity analysis of MVQM is discussed in Section 5.1.3.3.

### 5.4.3. Simulation Results and Performance Analysis

To evaluate the performance of the proposed CQ CBR algorithms, Q2 (frame-level control) and TM5 (macroblock-level control) are used as reference algorithms. Four test sequences (table tennis, football, flower garden, and mobile calendar) of 150 frames in length and 30 frames/second are used in the simulations. The color format is YUV 4:2:0. The first three sequences have frame size 352x240, and the last one has frame size 352x288. Each sequence is MPEG-4 encoded using two GOV structures (IPPP with intra period of 30 frames, and IBBP with intra period of 15 frames) at 1000, 2000, 3000, and 4000 kbps. The buffer sizes are set to be 0.5 second (15 frames) of the target bitrate. Frame skipping is disabled in all simulations. Default motion compensation search range (16 pixels) is used.

For FLCQ DTQ and MVCQ DTQ, because $\lambda$ is used to control how constant the quality is, $\lambda$ is swept from 0 to 1 (exclusive) to evaluate its effect. A total of 29 different $\lambda$s (0.00, 0.01, 0.02 ... 0.10, 0.15, 0.20 ... 0.95, and 0.999) are simulated. Due to the large amount of data, only the critical portions of the simulation results are shown in Table 5.6 (IPPP) and Table 5.7 (IBBP). More details can be found in the attached CD. Four sample figures illustrating the effect of $\lambda$ are shown in Figure 5.8 to Figure 5.11. In each figure, the top three subplots show the effect of $\lambda$ over the average PSNR, PSNR variance, and bitrate. The bottom subplot shows the average PSNR versus the PSNR variance. Since the goal is to have high PSNR while having low PSNR variance, the lower-right corner of this subplot is the best area. Since $\lambda$ is swept from 0 to 1, the starting point of the curve ($\lambda$=0) is marked with a circle for FLCQ and a square for

MVCQ. The solid curves show how the average PSNR and PSNR variance are affected as $\lambda$ changes from 0 to 1.

To make a fair comparison, the frame-level QP control algorithms (Q2 and FLCQ) and macroblock-level QP control algorithms (TM5 and MVCQ) are compared separately. In general, as demonstrated in the results, within each category, the proposed algorithms provide a similar or higher PSNR while having lower PSNR variance than the reference algorithms. For some instances, (close to) zero PSNR variance is achieved.

Note that the frame-level QP control algorithms tend to have higher average PSNR and higher PSNR variance than the macroblock-level QP control algorithms. Larger PSNR variance in frame-level QP control algorithms is expected since they allow only one QP to control the quality. Thus, the frame-level QP control cannot meet the target quality as well as the macroblock QP control. On the other hand, because of the extra degree of freedom in macroblock QP control, extra headers are inserted to keep track of the QP changes. Thus, the macroblock QP control spends bits less efficiently than the frame-level QP control. Note that the same phenomenon has been observed in the previous section.

As shown in the tables and figures, as $\lambda$ increases from 0 to (nearly) 1, the actual bitrate meets the target bitrate more accurately, the average PSNR increases, but the PSNR variance increases. For most cases, the lowest PSNR variance occurs when $\lambda$ is 0, which is expected from (56). Also, as $\lambda$ increases, the dominant term in (56) shifts from low PSNR variance to high PSNR and accurate bitrate. This phenomenon is also clearly shown in the results. Generally speaking, $\lambda$=0.15 seems to give good compromise among accuracy to the target bitrate, high PSNR, and low PSNR variance.

As a final remark, note that for cases where the PSNR variance is (close to) zero, the results appear to be independent of $\lambda$. This is due to the way the buffer is controlled. As shown in (55), when buffer occupancy ($b$) is between $b_{Low}$ (20%) and $b_{High}$ (70%), $S(b)$ is zero. This in turn zeros out the first term of (56) which tries to maximize the PSNR. Hence, the target PSNR is never being changed, and $\lambda$ has no effect on the results. For all simulation results that are independent of $\lambda$, the buffer occupancy is always between 20% and 70%.

Table 5.6: Selected Simulation Results for IPPP GOV Structure
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Target Bitrate: 1000 kbps** | | | | | | | | | | | | | |
| **Q2** | – | 1042 | 996 | 1001 | 1003 | 34.34 | 30.53 | 25.83 | 26.77 | 6.68 | 0.86 | 0.97 | 0.20 |
| **TM5** | – | 1005 | 999 | 1001 | 1000 | 33.30 | 30.36 | 25.14 | 26.31 | 5.83 | 0.80 | 0.87 | 0.15 |
| | 0.00 | 638 | 882 | 985 | 911 | 31.88 | 29.96 | 25.76 | 26.36 | 0.26 | 0.02 | 1.29 | 0.01 |
| | 0.05 | 820 | 982 | 982 | 967 | 33.23 | 30.39 | 25.76 | 26.63 | 2.04 | 0.28 | 1.04 | 0.03 |
| **FLCQ** | 0.10 | 969 | 965 | 975 | 969 | 34.01 | 30.25 | 25.73 | 26.64 | 4.54 | 0.62 | 1.01 | 0.03 |
| **DTQ** | 0.15 | 997 | 961 | 972 | 973 | 34.11 | 30.25 | 25.72 | 26.65 | 5.18 | 0.62 | 0.98 | 0.04 |
| | 0.20 | 1018 | 961 | 970 | 971 | 34.26 | 30.24 | 25.72 | 26.65 | 6.09 | 0.65 | 0.98 | 0.04 |
| | 0.999 | 1009 | 966 | 972 | 977 | 34.04 | 30.27 | 25.75 | 26.67 | 6.16 | 0.70 | 0.99 | 0.04 |
| | 0.00 | 634 | 919 | 989 | 992 | 31.66 | 30.14 | 25.73 | 26.65 | 0.32 | 0.01 | 1.19 | 0.00 |
| | 0.05 | 847 | 998 | 984 | 992 | 33.20 | 30.47 | 25.71 | 26.65 | 2.28 | 0.11 | 1.04 | 0.00 |
| **MVCQ** | 0.10 | 992 | 982 | 980 | 992 | 33.90 | 30.36 | 25.70 | 26.65 | 4.36 | 0.21 | 0.93 | 0.00 |
| **DTQ** | 0.15 | 1013 | 972 | 978 | 992 | 34.03 | 30.30 | 25.71 | 26.65 | 4.84 | 0.30 | 0.88 | 0.00 |
| | 0.20 | 1012 | 970 | 976 | 992 | 34.04 | 30.29 | 25.70 | 26.65 | 4.95 | 0.30 | 0.91 | 0.00 |
| | 0.999 | 1019 | 968 | 978 | 992 | 34.13 | 30.32 | 25.71 | 26.65 | 8.23 | 0.76 | 0.96 | 0.00 |
| **Target Bitrate: 2000 kbps** | | | | | | | | | | | | | |
| **Q2** | – | 1858 | 1990 | 2002 | 1999 | 37.05 | 33.96 | 29.58 | 30.26 | 3.40 | 1.04 | 1.22 | 0.21 |
| **TM5** | – | 2001 | 1999 | 2000 | 2000 | 36.63 | 33.51 | 28.74 | 29.58 | 4.92 | 0.76 | 1.24 | 0.17 |
| | 0.00 | 1454 | 2001 | 1957 | 2016 | 35.76 | 33.94 | 29.52 | 30.26 | 0.34 | 0.42 | 0.07 | 0.79 |
| | 0.05 | 1584 | 1879 | 2046 | 1978 | 36.34 | 33.62 | 29.73 | 30.21 | 1.39 | 0.47 | 0.21 | 0.33 |
| **FLCQ** | 0.10 | 1940 | 1896 | 2013 | 1990 | 37.29 | 33.69 | 29.62 | 30.25 | 5.34 | 0.48 | 0.50 | 0.25 |
| **DTQ** | 0.15 | 1993 | 1911 | 2003 | 2003 | 37.47 | 33.74 | 29.59 | 30.29 | 6.38 | 0.34 | 0.52 | 0.20 |
| | 0.20 | 2015 | 1909 | 1994 | 2007 | 37.53 | 33.73 | 29.55 | 30.31 | 6.79 | 0.56 | 0.60 | 0.18 |
| | 0.999 | 2020 | 1967 | 1966 | 1989 | 37.64 | 33.87 | 29.48 | 30.25 | 8.20 | 0.47 | 1.03 | 0.36 |
| | 0.00 | 1434 | 1990 | 1866 | 2015 | 35.54 | 33.80 | 29.14 | 30.21 | 0.09 | 0.18 | 0.00 | 0.54 |
| | 0.05 | 1687 | 1942 | 2041 | 1987 | 36.53 | 33.67 | 29.62 | 30.19 | 1.92 | 0.27 | 0.29 | 0.25 |
| **MVCQ** | 0.10 | 1916 | 1921 | 2009 | 1988 | 37.26 | 33.62 | 29.52 | 30.19 | 5.58 | 0.30 | 0.60 | 0.21 |
| **DTQ** | 0.15 | 1993 | 1918 | 1992 | 1989 | 37.53 | 33.62 | 29.48 | 30.19 | 7.08 | 0.31 | 0.76 | 0.22 |
| | 0.20 | 2013 | 1920 | 1979 | 1992 | 37.61 | 33.62 | 29.44 | 30.19 | 7.54 | 0.31 | 0.91 | 0.21 |
| | 0.999 | 2011 | 1930 | 1965 | 1992 | 37.57 | 33.67 | 29.40 | 30.20 | 7.88 | 0.39 | 1.03 | 0.40 |
| **Target Bitrate: 4000 kbps** | | | | | | | | | | | | | |
| **Q2** | – | 2754 | 3963 | 4012 | 4008 | 39.39 | 37.74 | 34.57 | 34.82 | 0.19 | 2.13 | 1.89 | 0.37 |
| **TM5** | – | 3995 | 3999 | 4001 | 4000 | 39.94 | 37.04 | 33.39 | 33.65 | 2.31 | 0.73 | 1.55 | 0.26 |
| | 0.00 | 2761 | 3968 | 3956 | 4093 | 39.32 | 37.69 | 34.67 | 35.01 | 0.12 | 1.48 | 0.04 | 0.02 |
| | 0.05 | 2778 | 3741 | 3960 | 4092 | 39.44 | 37.37 | 34.68 | 35.01 | 0.12 | 1.35 | 0.04 | 0.02 |
| **FLCQ** | 0.10 | 3971 | 3861 | 3960 | 4082 | 41.45 | 37.57 | 34.68 | 34.99 | 5.69 | 1.79 | 0.04 | 0.06 |
| **DTQ** | 0.15 | 3987 | 3835 | 3975 | 4072 | 41.41 | 37.55 | 34.70 | 34.97 | 5.97 | 1.71 | 0.06 | 0.10 |
| | 0.20 | 3927 | 3833 | 3993 | 4071 | 41.36 | 37.55 | 34.72 | 34.96 | 5.56 | 1.49 | 0.10 | 0.10 |
| | 0.999 | 3935 | 3893 | 3998 | 4057 | 41.33 | 37.61 | 34.73 | 34.93 | 6.15 | 1.44 | 0.11 | 0.25 |
| | 0.00 | 2710 | 3878 | 3975 | 4088 | 39.22 | 37.46 | 34.70 | 34.95 | 0.07 | 1.04 | 0.03 | 0.02 |
| | 0.05 | 3295 | 3782 | 3980 | 4089 | 40.61 | 37.32 | 34.71 | 34.92 | 1.88 | 0.78 | 0.03 | 0.03 |
| **MVCQ** | 0.10 | 3849 | 3798 | 3987 | 4080 | 41.11 | 37.36 | 34.72 | 34.91 | 2.84 | 0.75 | 0.03 | 0.04 |
| **DTQ** | 0.15 | 3915 | 3818 | 3995 | 4072 | 41.13 | 37.37 | 34.73 | 34.89 | 2.94 | 0.58 | 0.03 | 0.05 |
| | 0.20 | 3919 | 3891 | 3998 | 4066 | 41.13 | 37.49 | 34.73 | 34.88 | 2.94 | 0.38 | 0.04 | 0.07 |
| | 0.999 | 3931 | 3863 | 4011 | 4049 | 41.24 | 37.44 | 34.75 | 34.84 | 5.70 | 0.60 | 0.06 | 0.22 |

Table 5.7: Selected Simulation Results for IBBP GOV Structure
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Target Bitrate: 1000 kbps** | | | | | | | | | | | | | |
| **Q2** | – | 1016 | 1002 | 1005 | 1006 | 32.94 | 29.74 | 24.67 | 26.35 | 3.07 | 1.18 | 1.68 | 0.75 |
| **TM5** | – | 1000 | 1000 | 1001 | 1000 | 33.76 | 30.25 | 26.11 | 27.88 | 4.26 | 2.51 | 0.91 | 0.50 |
| | 0.00 | 587 | 852 | 892 | 949 | 31.31 | 29.10 | 25.24 | 27.27 | 0.46 | 0.53 | 1.23 | 0.01 |
| | 0.05 | 826 | 937 | 950 | 972 | 32.89 | 29.67 | 25.56 | 27.43 | 2.23 | 0.56 | 1.01 | 0.02 |
| **FLCQ** | 0.10 | 954 | 950 | 1004 | 977 | 33.66 | 29.87 | 25.94 | 27.48 | 2.99 | 0.23 | 0.94 | 0.03 |
| **DTQ** | 0.15 | 964 | 957 | 987 | 979 | 33.49 | 29.90 | 26.11 | 27.47 | 2.83 | 0.22 | 0.55 | 0.03 |
| | 0.20 | 965 | 979 | 983 | 984 | 33.75 | 30.01 | 26.19 | 27.50 | 3.16 | 0.20 | 0.63 | 0.04 |
| | 0.999 | 965 | 973 | 982 | 1003 | 33.67 | 30.00 | 26.29 | 27.58 | 3.60 | 0.20 | 0.74 | 0.12 |
| | 0.00 | 521 | 861 | 890 | 981 | 30.17 | 29.12 | 25.11 | 27.39 | 1.05 | 0.53 | 1.24 | 0.00 |
| | 0.05 | 759 | 936 | 955 | 983 | 32.30 | 29.65 | 25.58 | 27.42 | 3.37 | 0.45 | 0.98 | 0.01 |
| **MVCQ** | 0.10 | 933 | 951 | 1001 | 981 | 33.36 | 29.89 | 25.91 | 27.42 | 4.71 | 0.16 | 0.81 | 0.01 |
| **DTQ** | 0.15 | 964 | 966 | 986 | 980 | 33.69 | 29.96 | 26.19 | 27.43 | 3.48 | 0.11 | 0.52 | 0.01 |
| | 0.20 | 964 | 971 | 983 | 981 | 33.70 | 30.00 | 26.22 | 27.44 | 3.16 | 0.09 | 0.60 | 0.01 |
| | 0.999 | 970 | 978 | 982 | 1010 | 33.66 | 30.04 | 26.25 | 27.65 | 3.97 | 0.12 | 0.76 | 0.19 |
| **Target Bitrate: 2000 kbps** | | | | | | | | | | | | | |
| **Q2** | – | 1952 | 2002 | 2007 | 2009 | 36.41 | 33.16 | 28.85 | 30.44 | 4.42 | 1.52 | 2.15 | 0.75 |
| **TM5** | – | 2001 | 2000 | 2001 | 2001 | 36.84 | 33.33 | 29.61 | 30.93 | 4.55 | 3.23 | 1.90 | 0.98 |
| | 0.00 | 1443 | 1910 | 1874 | 1959 | 35.38 | 33.07 | 29.04 | 30.63 | 0.41 | 0.55 | 1.86 | 0.94 |
| | 0.05 | 1595 | 1924 | 1916 | 2003 | 36.01 | 33.30 | 29.25 | 31.09 | 1.55 | 0.32 | 1.80 | 0.32 |
| **FLCQ** | 0.10 | 1883 | 1926 | 1946 | 1999 | 36.88 | 33.38 | 29.43 | 31.07 | 3.83 | 0.26 | 1.41 | 0.28 |
| **DTQ** | 0.15 | 1925 | 1956 | 1980 | 2002 | 37.32 | 33.45 | 29.66 | 31.06 | 5.98 | 0.24 | 1.18 | 0.28 |
| | 0.20 | 1925 | 1971 | 1980 | 2000 | 37.25 | 33.49 | 29.67 | 31.06 | 5.42 | 0.24 | 1.18 | 0.29 |
| | 0.999 | 1938 | 1960 | 1971 | 1990 | 37.08 | 33.45 | 29.85 | 31.02 | 4.90 | 0.20 | 1.25 | 0.62 |
| | 0.00 | 1368 | 1936 | 1872 | 1947 | 35.03 | 33.31 | 28.96 | 30.67 | 0.23 | 0.02 | 1.90 | 0.88 |
| | 0.05 | 1692 | 1936 | 1947 | 2006 | 36.39 | 33.31 | 29.41 | 31.06 | 2.30 | 0.02 | 1.19 | 0.31 |
| **MVCQ** | 0.10 | 1887 | 1936 | 1998 | 1997 | 37.07 | 33.31 | 29.85 | 31.05 | 4.91 | 0.02 | 0.77 | 0.29 |
| | 0.15 | 1968 | 1936 | 1987 | 1999 | 37.26 | 33.31 | 29.85 | 31.04 | 5.90 | 0.02 | 0.77 | 0.28 |
| | 0.20 | 1937 | 1936 | 1988 | 1993 | 37.19 | 33.31 | 29.86 | 31.03 | 5.47 | 0.02 | 0.75 | 0.31 |
| | 0.999 | 1936 | 1936 | 1971 | 1980 | 37.14 | 33.31 | 29.84 | 31.00 | 6.18 | 0.02 | 1.13 | 0.61 |
| **Target Bitrate: 4000 kbps** | | | | | | | | | | | | | |
| **Q2** | – | 2868 | 4034 | 4007 | 4008 | 39.46 | 37.39 | 34.02 | 35.07 | 0.26 | 3.23 | 3.15 | 1.25 |
| **TM5** | – | 4000 | 4000 | 4001 | 4001 | 39.93 | 36.76 | 33.69 | 34.59 | 4.06 | 4.24 | 3.58 | 2.17 |
| | 0.00 | 2880 | 3772 | 4031 | 3927 | 39.32 | 37.01 | 34.47 | 35.28 | 0.13 | 1.36 | 3.52 | 1.64 |
| | 0.05 | 2887 | 3731 | 4088 | 3912 | 39.56 | 37.06 | 34.82 | 35.28 | 0.24 | 1.29 | 2.30 | 1.61 |
| **FLCQ** | 0.10 | 3864 | 3831 | 4038 | 3900 | 41.26 | 37.33 | 34.78 | 35.34 | 4.77 | 1.65 | 2.33 | 1.39 |
| **DTQ** | 0.15 | 3828 | 3845 | 4033 | 3884 | 41.14 | 37.38 | 34.77 | 35.30 | 3.72 | 1.67 | 2.30 | 1.42 |
| | 0.20 | 3886 | 3849 | 4037 | 3871 | 41.06 | 37.37 | 34.80 | 35.29 | 4.72 | 1.64 | 2.22 | 1.40 |
| | 0.999 | 3864 | 3867 | 4025 | 3873 | 41.39 | 37.45 | 34.89 | 35.39 | 7.39 | 1.78 | 1.88 | 1.15 |
| | 0.00 | 2830 | 3573 | 4051 | 4045 | 39.33 | 36.72 | 34.37 | 35.44 | 0.07 | 1.29 | 2.83 | 0.85 |
| | 0.05 | 3370 | 3796 | 4019 | 3962 | 40.57 | 37.10 | 34.63 | 35.47 | 1.89 | 1.36 | 1.83 | 0.36 |
| **MVCQ** | 0.10 | 3751 | 3846 | 4038 | 3943 | 41.01 | 37.23 | 34.77 | 35.44 | 2.70 | 1.21 | 1.00 | 0.34 |
| **DTQ** | 0.15 | 3844 | 3848 | 4004 | 3927 | 41.17 | 37.23 | 34.74 | 35.42 | 3.73 | 1.05 | 1.06 | 0.32 |
| | 0.20 | 3860 | 3851 | 4002 | 3926 | 41.19 | 37.22 | 34.74 | 35.42 | 3.91 | 1.01 | 1.05 | 0.32 |
| | 0.999 | 3907 | 3869 | 4027 | 3979 | 41.25 | 37.28 | 34.81 | 35.46 | 5.28 | 1.36 | 1.23 | 0.71 |

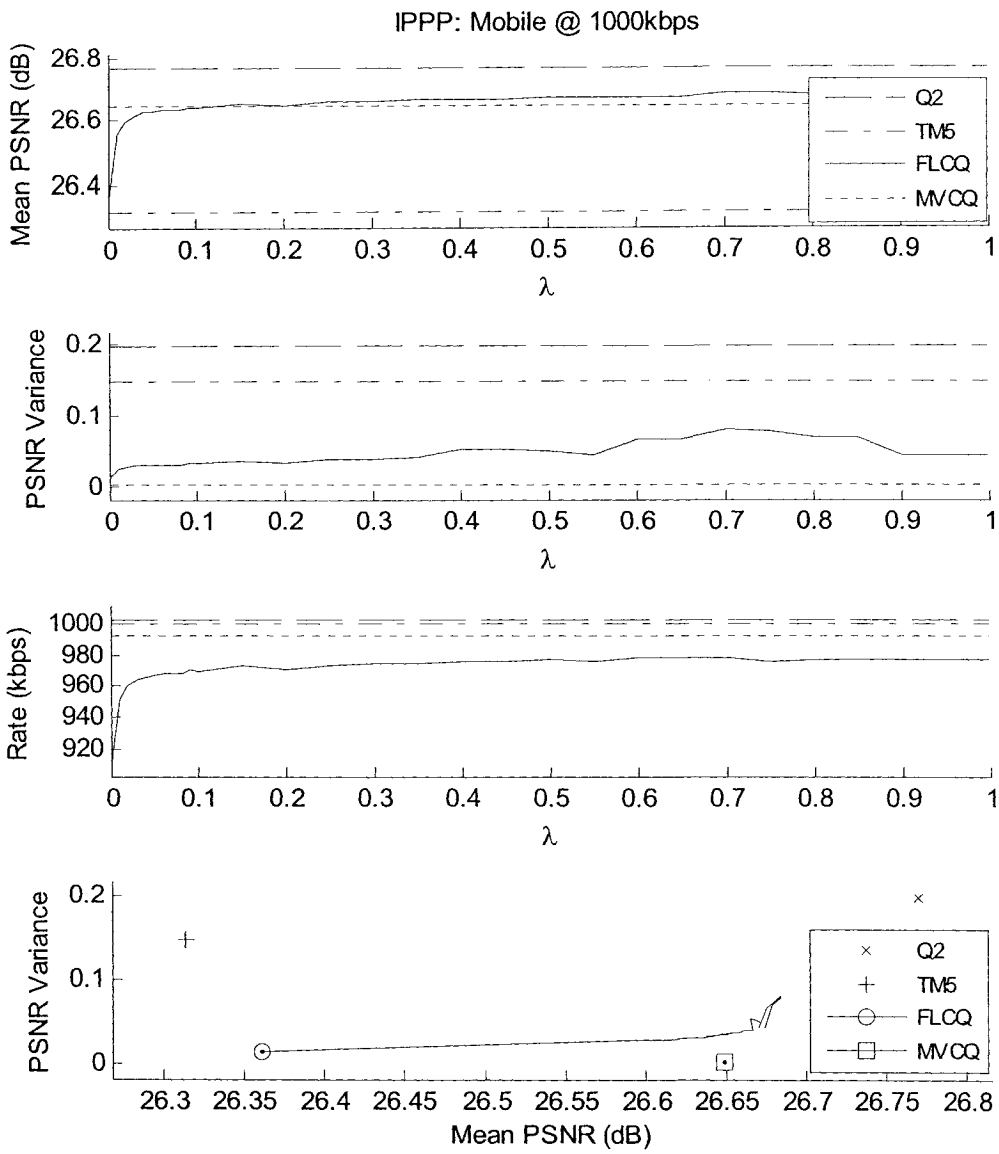



Figure 5.8: Effect of $\lambda$ for Mobile Calendar for Q2, TM5,
FLCQ DTQ, and MVCQ DTQ algorithms (GOV: IPPP, Rate: 1000 kbps)

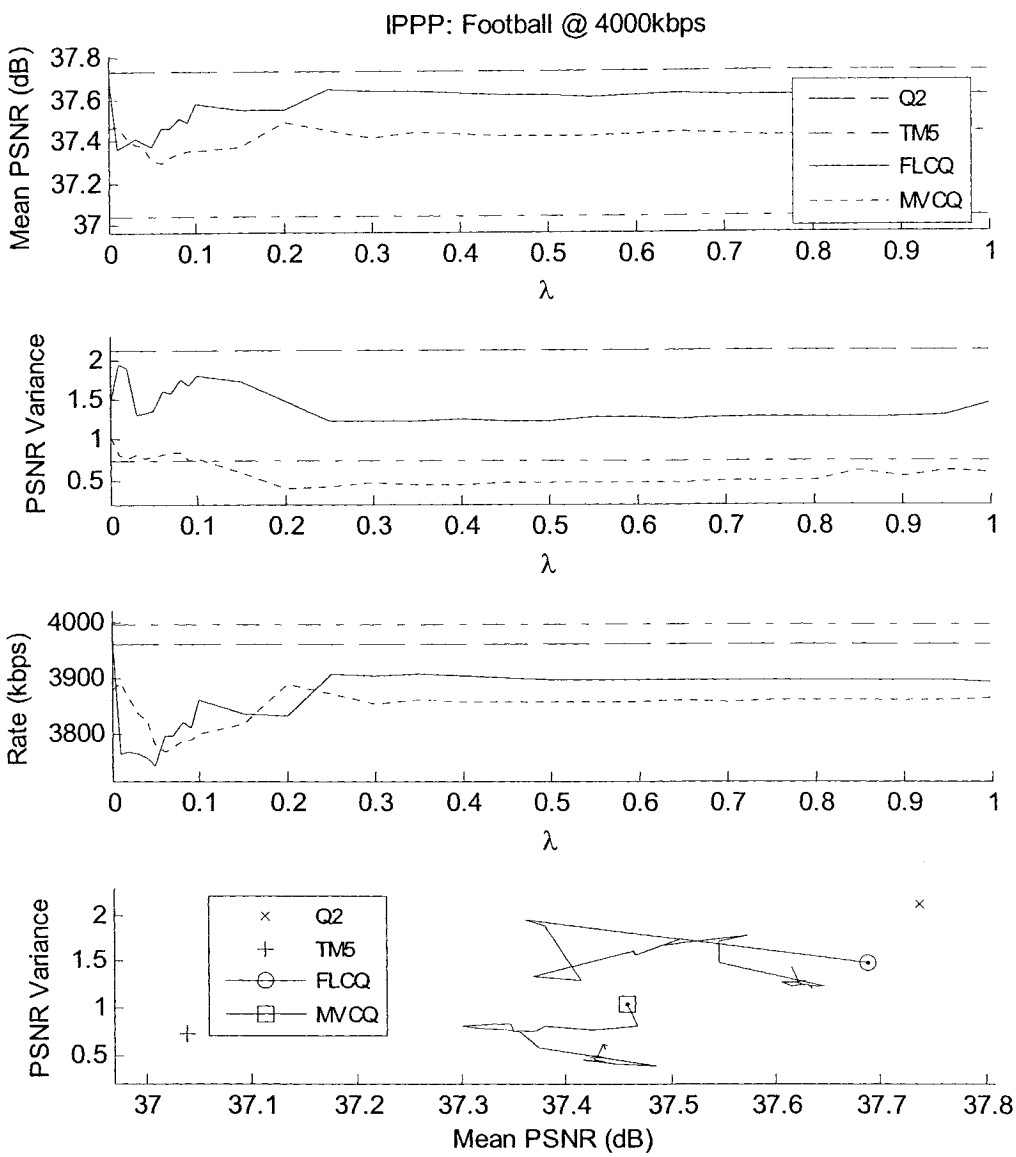



Figure 5.9: Effect of $\lambda$ for Football for Q2, TM5,
FLCQ DTQ, and MVCQ DTQ algorithms (GOV: IPPP, Rate: 4000 kbps)

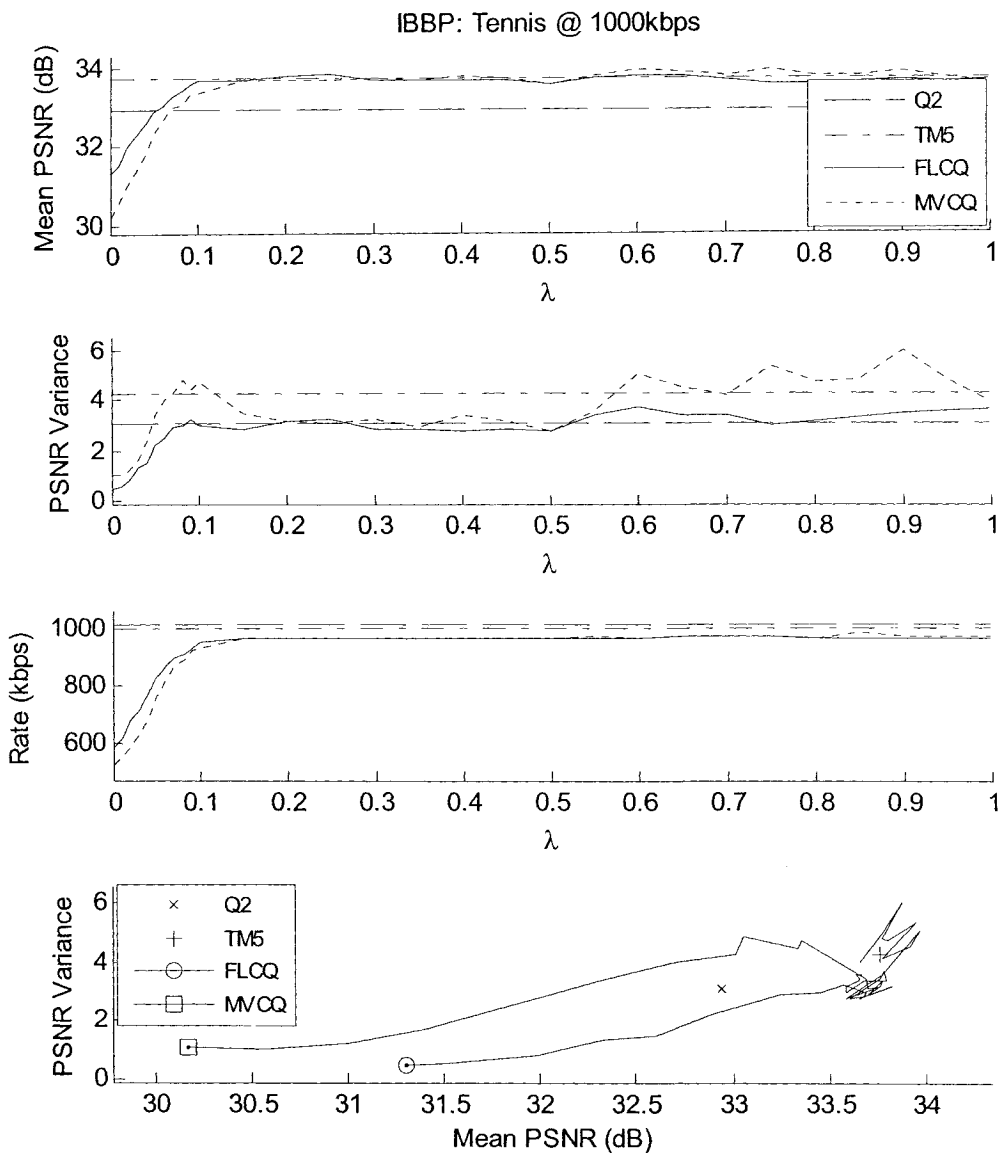



Figure 5.10: Effect of $\lambda$ for Table Tennis for Q2, TM5,
FLCQ DTQ, and MVCQ DTQ algorithms (GOV: IBBP, Rate: 1000 kbps)

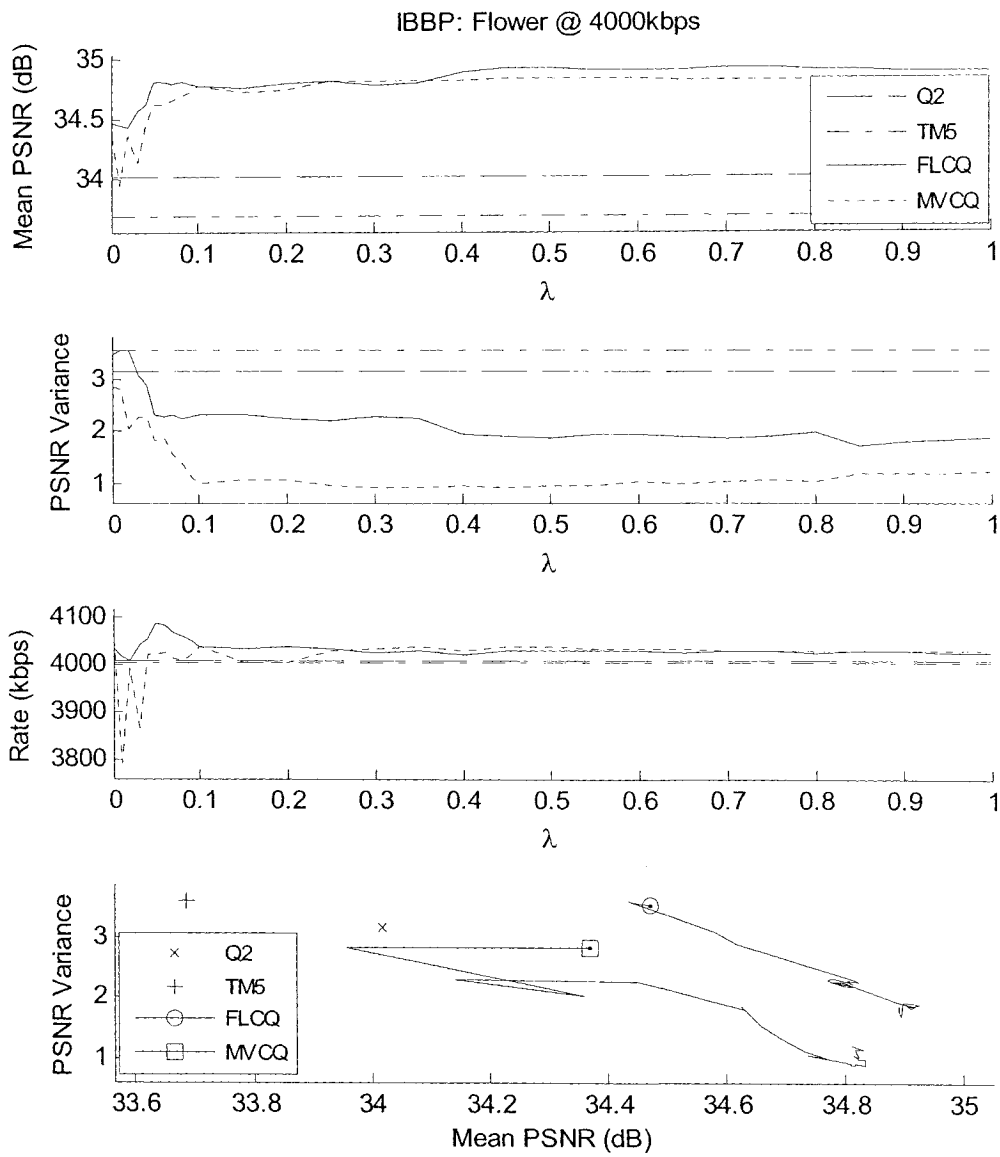



Figure 5.11: Effect of $\lambda$ for Flower Garden for Q2, TM5,
FLCQ DTQ, and MVCQ DTQ algorithms (GOV: IBBP, Rate: 4000 kbps)

## 5.5. Chapter Summary

In this chapter, three sets of constant quality control algorithms are proposed. Within each set, two algorithms are proposed that control the QP at either the frame-level or at the macroblock level. The key difference between the proposed CQ algorithms and the previously reported CQ algorithms is that the quality is controlled directly through the QM algorithm instead of indirectly through the target bit allocation. As demonstrated in the results, this approach provides accurate control in quality.

Although these algorithms all aim at reducing the quality variation, the exact goal is slightly different for each set. The only goal for the CQ VBR algorithms presented in Section 5.2 is to meet one target quality for the entire sequence. Since a buffer constraint is not considered, these algorithms are only constrained by the size of QP. For a reasonable and achievable target quality, these algorithms can meet the target with very little or no quality variation. Simulation results suggest that the performance of the FLCQ VBR algorithm approaches the OQP algorithm, while the MVCQ VBR algorithm gives even better performance than the OQP algorithm since it controls the quality at the MB level.

Two sets of CQ CBR algorithms are proposed in Section 5.3 and 5.4 – one set with fixed target quality and another set with dynamic target quality. Because of the buffer constraint, in general, the quality variations of these algorithms are higher than the VBR case. For both sets of algorithms, simulation results indicate that the proposed FLCQ/MVCQ algorithms outperform Q2/TM5 by offering similar or higher PSNR and lower PSNR variance. FLCQ tends to generate higher average PSNR but larger PSNR

variance than the MVCQ. For some instances, zero (very small) PSNR variance is achieved.

As discussed in Section 5.3, because setting a good target quality is not trivial, the algorithms proposed in Section 5.4 allows the target quality to be changed dynamically. An extra degree of freedom ($\lambda$) is introduced to balance the weighting among higher PSNR, lower PSNR variance, and the accuracy to the target bitrate. This extra degree of freedom allows a continuous transition of the optimization goal from low PSNR variance (purely CQ) to constant bitrate with high PSNR (as in the traditional CBR algorithms).

In this chapter, CQ control algorithms are designed for MPEG-4 encoding. In the next chapter, the concept of direct-quality-control used in this chapter will be extended to MPEG-4 transcoding. As will be shown in the next chapter, this approach is not only good for video encoding, but is also good for video transcoding.

# Chapter 6

# Constant-Quality CBR Rate-Control Algorithms for MPEG-4 Video Transcoding

In Chapter 5, constant quality (CQ) rate-control algorithms for MPEG-4 video encoding were studied. In this chapter, the method to reduce quality variations taken by these algorithms will be extended to CQ CBR (constant bit rate) control for MPEG-4 video transcoding. Similar to the two algorithms proposed in Section 5.4, the algorithms proposed in this chapter adjust the target quality dynamically at runtime. The transcoder architecture used in this chapter is the cascaded decoder-encoder pixel-domain architecture (see Figure 6.1) as mentioned in Section 2.3.1.

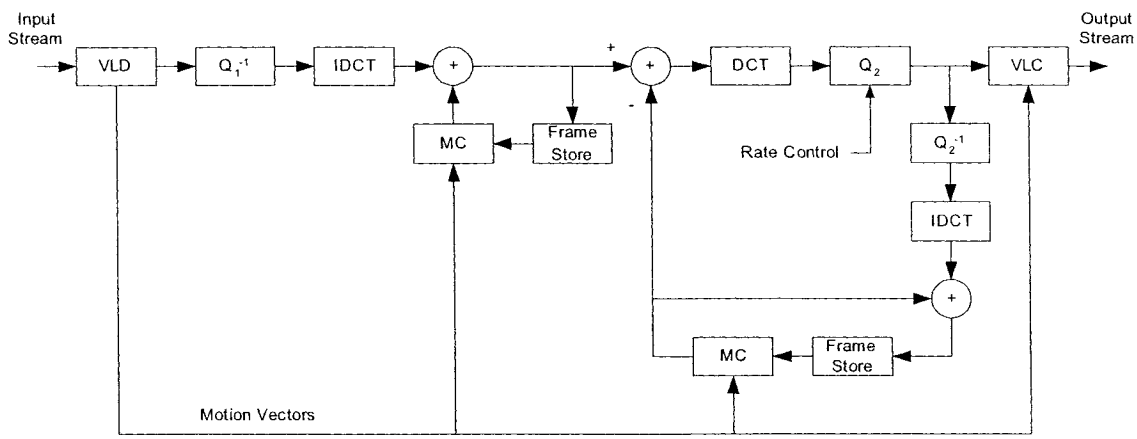



Figure 6.1: Cascaded decoder-encoder pixel-domain transcoder architecture [72][76]

Although many rate control algorithms for video transcoding were proposed in the past, most of them aimed at CBR applications with the highest possible quality. The quality variation of the transcoded bitstream was not considered. There is only one rate-control algorithm reported in [117] that attempts to have a constant distortion (i.e., constant quality) transcoded bitstream. It uses an open-loop transcoder architecture, and achieves constant quality by using frame dropping (FD) and coefficient dropping (CD) techniques. This means that the temporal resolution is changed. In this chapter, two constant-quality control algorithms are proposed that aim at reducing the bitrate, without changing neither temporal nor spatial resolution.

This chapter is organized as follows. Similar to the previous chapter, the quality-matching (QM) and bit-matching (BM) algorithms are presented in Section 6.1. The QM/BM algorithms compute the best quantization parameters (QPs) for a given target-quality/bit-budget. They are used by the two CQ CBR transcoding algorithms proposed in Section 6.2. These CQ CBR transcoding algorithms are developed based on the CQ CBR encoding algorithms presented in Section 5.4, and they adjust the target quality dynamically. Simulation results are presented and analyzed in Section 6.3. Section 6.4 summarizes the chapter.

## 6.1. Quality Matching and Bit Matching Algorithms for MPEG-4 Video Transcoding

Following a similar approach to that taken in Section 5.1, this section presents quality matching and bit matching algorithms for MPEG-4 video transcoding. They are

used to find the QP(s) such that the resulting compressed frame best matches a given target quality or bit budget. Note, since the transcoder lacks information about the original uncompressed video, the QM/BM algorithms presented in this section are not exactly the same as the ones presented in Section 5.1. Different methods are used to estimate the quality and number of bits.

Two types of QM/BM algorithms are presented in this section. The Frame-level Laplacian QM/BM Transcoding (FLQMT/FLBMT) algorithms are presented in Section 6.1.1 and 6.1.2. They use the MPEG-4 Laplacian distortion/rate model for transcoders developed in Section 3.2 to find the best QP for the entire frame. The Macroblock-level Viterbi QM/BM Transcoding (MVQMT/MVBMT) algorithms are presented in Section 6.1.3 and 6.1.4 respectively. Like their counterparts in video encoding, they use the Viterbi algorithm to determine QPs for each macroblock.

### 6.1.1. Frame-Level Laplacian Quality Matching Transcoding (FLQMT) Algorithm

The FLQMT algorithm is used to determine the frame QP that best matches the given target quality. The main difference between FLQMT and its counterpart FLQM proposed in Section 5.1.1 is that it uses the Laplacian distortion model for transcoding derived in Section 3.2.1 rather than the Laplacian distortion model for encoding. The FLQMT algorithm is described as follows:

**Step 1**: Use (36) to estimate the Laplacian parameters ($\alpha_{i,j}$) for DCT frequency ($i,j$) of the current frame where $i$=0..7, and $j$=0..7.

**Step 2**: The selected QP is the QP with the estimated MSE (mean squared error) closest to the target MSE ($MSE_{Target}$):

$$QP_{Selected} = \arg\min_{QP_2} \left| MSE_{Target} - \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8} MSE\left(QP_1, QP_2 \mid \alpha_{i,j}\right) \right| \tag{58}$$

where $QP_1$ is the average QP of the encoded input frame, $QP_2$ is the QP to be used at the transcoder ($QP_2$= 2, 4, 6... 62), and the *MSE* function gives the estimated MSE for DCT frequency $(i,j)$. It is derived in (24) and is defined as:

$$MSE\left(Q_1, Q_2 \mid \alpha\right) = \frac{2}{\alpha^2} + \sum_{j=1}^{\infty} e^{-\alpha S_j}\left[\left(T_j^2 - T_{j-1}^2\right) - 2\left(S_j + \frac{1}{\alpha}\right)\left(T_j - T_{j-1}\right)\right] \tag{59}$$

As mentioned in Section 3.2.1, not all frames/DCT coefficients will follow the Laplacian distribution. For the FLQM algorithm (for encoding), this is not a big issue since one can compensate for the deviation. However, the FLQMT can no longer compensate for the estimation error. This is because one cannot obtain the actual MSE of the transcoded frame. Computing the actual MSE requires the original frame, which is inaccessible at the transcoder. As a result, the FLQMT algorithm for transcoder is expected to be less accurate than its counterpart FLQM for encoding.

### 6.1.2. Frame-level Laplacian Bit Matching Transcoding (FLBMT) Algorithm

The FLBMT algorithm is used to determine the frame QP that best matches the given bit budget ($Bit_{Target}$). It is very similar to its counterpart FLBM (for encoding) presented in Section 5.1.2. The only difference is that it uses the Laplacian rate model for

transcoding (Section 3.2.2) rather than the Laplacian rate model for encoding (Section 3.1.3). The FLBMT algorithm is described as follows.

**Step 1**: Use (36) to estimate the Laplacian parameters ($\alpha_{i,j}$) for DCT frequency ($i,j$) of the current frame.

**Step 2**: The selected QP is the QP with the estimated number of bits that is closest to the target bits:

$$QP_{Selected} = \arg\min_{QP_2} \left| Bit_{Target} - \left( \varepsilon_{Bit} + \sum_{i=1}^{8}\sum_{j=1}^{8} Bit\left(QP_1, QP_2 \mid \alpha_{i,j}\right) \right) \right|$$
$$\varepsilon_{Bit} = ActualBit_{Previous} - \sum_{i=1}^{8}\sum_{j=1}^{8} Bit\left(PrevQP_1, PrevQP_2 \mid Prev\alpha_{i,j}\right) \quad (60)$$

where $QP$ = 2, 4, 6... 62, and $Bit(QP|\alpha_{i,j})$ is defined in (25), $ActualBit_{Previous}$, $PrevQP_1$, $PrevQP_2$, and $Prev\alpha$ are the actual number of bits, average input QP, transcoder QP, and Laplacian parameters of the previously coded frame of the same type.

Note that unlike the FLQMT algorithm presented in the previous section, the estimated number of bits *can* be compensated to increase its accuracy. This is because the actual number of bits of the transcoded bitstream is still available at the transcoder.

### 6.1.3. Macroblock-level Viterbi Quality Matching Transcoding (MVQMT) Algorithm

The MVQMT algorithm is very similar to its counterpart for encoding (MVQM) presented in Section 5.1.3. They both attempt to find the macroblock QPs that best match a given target quality. The main difference is that the encoding algorithm requires the actual MSE of each macroblock. However, this information is no longer available at the transcoder since the original video is not available. To handle this problem, the Laplacian distortion model for transcoding derived in Section 3.2.1 is used.

Let $X_U$ denote the random variable of the uncompressed samples, $X_I$ denotes random samples at the input of the transcoder, and $X_T$ denotes the transcoded samples at the output of the transcoder. The total MSE ($MSE_{Total}$) at the output of the transcoder relative to the original uncompressed sequence is

$$MSE_{Total} = E\left[\left(X_U - X_T\right)^2\right] \tag{61}$$

Equation (61) can also be expressed as

$$\begin{aligned} MSE_{Total} &= E\left[\left(X_U - X_T\right)^2\right] = E\left[\left(\left(X_U - X_I\right) + \left(X_I - X_T\right)\right)^2\right] \\ &= E\left[\left(X_U - X_I\right)^2 + \left(X_I - X_T\right)^2 + 2\left(X_U - X_I\right)\left(X_I - X_T\right)\right] \end{aligned} \tag{62}$$

or

$$MSE_{Total} = \underbrace{E\left[\left(X_U - X_I\right)^2\right]}_{A} + \underbrace{E\left[\left(X_I - X_T\right)^2\right]}_{B} + 2\underbrace{E\left[\underbrace{\left(X_U - X_I\right)}_{C1}\underbrace{\left(X_I - X_T\right)}_{C2}\right]}_{C} \tag{63}$$

Note that in equation (63), the first term ($A$) is the MSE presented at the input of the transcoder, the second term ($B$) is the MSE at the output of the transcoder relative to its input (not the original uncompressed input), and the third term ($C$) depends on the

correlation between the error at the input of the transcoder (*C1*) and the error introduced by the transcoder (*C2*). Note that only *B* and *C2* can be computed at the transcoder.

To further simplify (63), it is assumed that the error at the input of the transcoder (*C1*) and the error introduced by the transcoder (*C2*) are statistically independent. Note that in both encoding and transcoding, the main cause of error is the quantization error. For bitrate reduction transcoding, transcoder normally needs to use different quantization step sizes than the encoding step sizes. As a result, the assumption that the encoding quantization noise is independent of the transcoding quantization noise is generally true. With this assumption in mind, equation (63) can be expressed as:

$$MSE_{Total} \cong \underbrace{E\left[\left(X_U - X_I\right)^2\right]}_{A} + \underbrace{E\left[\left(X_I - X_T\right)^2\right]}_{B} + 2\underbrace{E\left[X_U - X_I\right]}_{D1}\underbrace{E\left[X_I - X_T\right]}_{D2} \qquad (64)$$

In general, the error at the input and the error introduced by the transcoder are introduced by quantization of DCT coefficients. If the input of the quantizer has zero mean and is symmetrically distributed about zero, the expected quantization error is also zero when a generalized uniform quantizer is used. As a result, the terms *D1* and *D2* in equation (64) are assumed to be zero. Thus, equation (64) becomes:

$$MSE_{Total} \cong MSE_{Input} + MSE_{Trans} \qquad (65)$$

where $MSE_{Input}$ (term *A*) is the MSE present at the transcoder input relative to the original uncompressed samples, and $MSE_{Trans}$ (term *B*) is the MSE introduced by the transcoder.

Since the MSE at the transcoder input is not available, in the MVQMT, $MSE_{Input}$ is estimated using the Laplacian distortion model, i.e.,

$$MSE_{Input} = \frac{1}{64}\sum_{i=1}^{8}\sum_{j=1}^{8} MSE\left(Q_1 \mid \alpha_{i,j}\right) \qquad (66)$$

where $Q_1$ is the quantization step size used at the input, and the *MSE* function gives the estimated MSE of DCT coefficient $(i,j)$, and is defined in (17). On the other hand, since all information required to compute $MSE_{Trans}$ is available, it is computed by using the actual MSE in the MVQMT algorithm.

With (65) in mind, the MVQMT is basically the same as the MVQM algorithm. The only difference is that it needs to deduct the input MSE from the target MSE. In other words, (44) should become:

$$\begin{aligned} MSE_{NewTarget} &= \frac{1}{M}\sum_{i=1}^{M} T_i \\ MSE_{NewTarget} &= MSE_{Target} - MSE_{Input} \end{aligned} \tag{67}$$

where $MSE_{NewTarget}$ is the target used by the MVQM algorithm for encoder, $MSE_{Target}$ is the overall MSE at the output of the transcoder relative to the original uncompressed input, and $MSE_{Input}$ is the estimated input MSE defined in (66).

### 6.1.4. Macroblock-level Viterbi Bit Matching Transcoding (MVBMT) Algorithm

The MVBMT algorithm for transcoding is almost the same as the MVBM algorithm for encoding defined in Section 5.1.4. There are only two differences:

- The MVBMT algorithm uses the MVQMT algorithm instead of using the MVQM algorithm.
- The MVBMT algorithm uses the Laplacian rate model for transcoding to estimate the number of bits instead of using the Laplacian rate model for encoding.

## 6.2. MPEG-4 Constant-Quality CBR Transcoding Rate-Control Algorithms with Dynamic Target Quality

Two CQ CBR algorithms for MPEG-4 video transcoding (FLCQT and MVCQT) are presented in this section. They are developed based on the FLCQ and MVCQ algorithms with dynamic target quality (DTQ) for MPEG-4 encoding developed in Section 5.4. The only difference is that they use the QM/BM algorithms for transcoding instead of QM/BM algorithms for encoding. This means that the FLCQT algorithm uses the FLQMT and FLBMT as the quality/bit matching algorithm, and the MVCQT algorithm uses the MVQMT/MVBMT as the quality/bit matching algorithm.

## 6.3. Simulation Results and Analysis

To evaluate the performance of the proposed transcoding CQ algorithms, four reference bitstreams are used for each target bitrate. Two reference bitstreams are encoded directly from the original sequence to the target bitrate using Q2 [7] (frame-level control) and TM5 (Test Model 5) [8] (macroblock-level control) as the rate control algorithms. The other two reference bitstreams are transcoded using Q2 and TM5 algorithms from the compressed bitstreams. Note that Q2 and TM5 were originally designed for video encoding. They are extended to transcoding in the simulations. To distinguish the transcoding Q2/TM5 with the encoding Q2/TM5, the Q2 and TM5 algorithms for transcoding is denoted as Q2T and TM5T in this chapter.

The input bitstreams of the transcoder are TM5 encoded at 4 Mbps using two GOV) structures (IPPP with intra period of 30 frames, and IBBP with intra period of 15

frames). The input bitstream are transcoded to 1, 2, and 3 Mbps with the same GOV structure as the input. Four input sequences are used: table tennis (352x240 300 frames), football (352x240 210 frames), flower garden (352x240 150 frames), and mobile calendar (352x288 300 frames). They all have frame rate of 30 frame/second and color format as YUV 4:2:0. The buffer sizes are set to be 0.5 second (15 frames) of the target bitrate. Frame skipping is disabled in all simulations.

Since the behaviour of the transcoder FLCQT and MVCQT algorithm depends on $\lambda$, $\lambda$ is swept from 0 to 1 (exclusive). A total of 19 different $\lambda$s (0.00, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.08, 0.10, 0.12, 0.14, 0.15, 0.16, 0.18, 0.20, 0.40, 0.60, 0.80, and 0.999) are simulated. Simulation results for IPPP GOV structure are listed in Table 6.1 to Table 6.3. Simulation results for IBBP GOV structure are listed in Table 6.4 to Table 6.6. Three sample figures that illustrate the effect of $\lambda$ are shown in Figure 6.2 to Figure 6.4. Two sample figures that illustrate the actual and the estimated PSNR (peak signal-to-noise ratio) at the output of transcoder are shown in Figure 6.5 and Figure 6.6.

As shown in the tables, when comparing the results of transcoding with direct encoding, a general observation is that the average PSNRs of the transcoded bitstreams are about 1~2 dB lower than those that are directly encoded. One reason for this drop is that the transcoders do not perform motion estimation again. Thus, motion vectors reused in transcoding may not be as good as recalculated motion vectors. Another cause of the quality degradation is the requantization error introduced by the second quantization. However, despite having a slightly lower average PSNR, the transcoded bitstreams (especially the ones using the proposed algorithms) generally give a lower quality variation than bitstreams that are directly encoded by the reference algorithms.

When the proposed transcoding algorithms (FLCQT and MVCQT) are compared with the transcoding reference algorithms (Q2T and TM5T), it can be observed that the proposed algorithms provide lower PSNR variance while having similar average PSNR and lower bitrate than the reference algorithms. This normally happens when $\lambda$ is small ($\lambda$<0.10 for instance). Note that the proposed algorithms generally have better performance than the reference algorithms for the IBBP GOV structures. For IPPP structures, the proposed algorithms give similar performance to the reference algorithms. However, it should be noted that the proposed algorithms still offer a way to control the quality variations through $\lambda$. This extra degree of control does not exist in the reference algorithms.

As shown in the tables and figures, as $\lambda$ increases from 0 to (nearly) 1, the actual bitrate meets the target bitrate more accurately, the average PSNR increases, but the PSNR variance increases. For most cases, the lowest PSNR variance occurs when $\lambda$ is 0. This is expected since the target PSNR is determined by using the cost function defined in equation (56). Also, as $\lambda$ increases, the dominant term in the cost function shifts from low PSNR variance to high PSNR and accurate bitrate. This phenomenon is also clearly shown in the results. Generally speaking, $\lambda$=0.10 seems to give good compromise among accuracy to the target bitrate, high PSNR, and low PSNR variance.

As a final observation, in Figure 6.5 and Figure 6.6, the actual and the estimated PSNR for two sequences are shown. It can be observed that the prediction made by the FLCQT is less accurate than the one made by the MVCQT. This is expected since the estimation made by FLCQT is based only on the Laplacian parameters estimated from the quantized DCT coefficients. When there is a fewer number of non-zero DCT

coefficients, or when input coefficients do not follow the Laplacian distribution, the estimation made by the Laplacian rate/distortion model is expected to be less accurate. On the other hand, the estimation PSNR made by MVCQT follows more closely to the actual PSNR. The reason for this is that the MSE used by the MVCQT is obtained from both the estimated the input MSE (using the Laplacian distortion model) and the actual MSE introduced by the transcoder. Since actual MSE is used as part of the estimation, it is expected that the MVCQT algorithm gives a more accurate prediction than the FLCQT algorithm.

Table 6.1: Simulation Results for IPPP GOV Structure @ 1000 kbps
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Q2** | – | 1057 | 1014 | 1003 | 1002 | 32.88 | 31.92 | 25.90 | 27.07 | 7.74 | 6.65 | 0.95 | 0.61 |
| **TM5** | – | 1001 | 1000 | 1000 | 1001 | 31.84 | 31.70 | 25.18 | 26.70 | 7.21 | 6.54 | 0.87 | 0.62 |
| **Q2T** | – | 1045 | 1007 | 1000 | 1002 | 32.27 | 31.54 | 25.15 | 26.41 | 6.99 | 6.46 | 1.23 | 0.50 |
| **TM5T** | – | 1000 | 1000 | 1001 | 1000 | 31.29 | 31.38 | 24.47 | 26.00 | 7.57 | 6.58 | 1.11 | 0.61 |
| | 0.00 | 786 | 812 | 965 | 797 | 30.78 | 30.38 | 24.96 | 25.45 | 0.94 | 0.71 | 0.95 | 0.12 |
| | 0.01 | 799 | 844 | 965 | 875 | 30.87 | 30.54 | 24.96 | 25.85 | 2.36 | 0.86 | 0.95 | 0.16 |
| | 0.02 | 836 | 862 | 967 | 927 | 31.12 | 30.64 | 24.97 | 26.09 | 3.51 | 1.12 | 0.95 | 0.21 |
| | 0.03 | 863 | 870 | 967 | 952 | 31.31 | 30.69 | 24.97 | 26.20 | 4.04 | 1.22 | 0.94 | 0.29 |
| | 0.04 | 872 | 883 | 966 | 978 | 31.42 | 30.73 | 24.96 | 26.31 | 4.62 | 1.49 | 0.95 | 0.36 |
| | 0.05 | 887 | 901 | 964 | 963 | 31.50 | 30.81 | 24.95 | 26.25 | 4.99 | 1.77 | 0.97 | 0.32 |
| | 0.06 | 911 | 905 | 964 | 972 | 31.64 | 30.77 | 24.95 | 26.28 | 5.74 | 2.17 | 0.97 | 0.36 |
| | 0.08 | 935 | 927 | 962 | 977 | 31.73 | 30.85 | 24.95 | 26.29 | 5.98 | 2.77 | 0.97 | 0.44 |
| | 0.10 | 948 | 937 | 965 | 975 | 31.76 | 30.95 | 24.96 | 26.26 | 6.21 | 3.11 | 0.95 | 0.39 |
| **FLCQT** | 0.12 | 966 | 946 | 966 | 978 | 31.87 | 31.00 | 24.97 | 26.28 | 6.91 | 3.46 | 0.95 | 0.43 |
| | 0.14 | 984 | 954 | 963 | 978 | 31.91 | 31.06 | 24.96 | 26.28 | 7.14 | 3.76 | 0.96 | 0.58 |
| | 0.15 | 980 | 961 | 963 | 977 | 31.94 | 31.10 | 24.96 | 26.29 | 7.33 | 3.95 | 0.96 | 0.59 |
| | 0.16 | 980 | 962 | 963 | 978 | 31.91 | 31.12 | 24.96 | 26.29 | 7.35 | 4.00 | 0.96 | 0.60 |
| | 0.18 | 982 | 972 | 967 | 981 | 31.88 | 31.18 | 24.97 | 26.29 | 6.89 | 4.40 | 0.94 | 0.45 |
| | 0.20 | 985 | 980 | 967 | 982 | 31.97 | 31.23 | 24.97 | 26.33 | 7.81 | 4.69 | 0.94 | 0.65 |
| | 0.40 | 990 | 1006 | 969 | 984 | 32.00 | 31.49 | 24.97 | 26.33 | 8.19 | 7.13 | 0.96 | 0.48 |
| | 0.60 | 998 | 1017 | 971 | 985 | 31.98 | 31.53 | 24.98 | 26.33 | 7.66 | 7.87 | 0.99 | 0.66 |
| | 0.80 | 999 | 1017 | 982 | 987 | 31.86 | 31.53 | 25.00 | 26.32 | 6.73 | 7.66 | 1.04 | 0.50 |
| | 0.999 | 1002 | 1015 | 982 | 985 | 31.78 | 31.40 | 24.99 | 26.31 | 6.83 | 6.67 | 0.99 | 0.57 |
| | 0.00 | 820 | 858 | 1006 | 879 | 30.84 | 30.50 | 25.02 | 25.73 | 0.87 | 0.65 | 1.55 | 0.31 |
| | 0.01 | 847 | 867 | 998 | 909 | 31.04 | 30.58 | 25.00 | 25.90 | 1.37 | 0.72 | 1.55 | 0.28 |
| | 0.02 | 861 | 875 | 995 | 944 | 31.18 | 30.63 | 25.01 | 26.08 | 2.35 | 0.91 | 1.46 | 0.30 |
| | 0.03 | 873 | 892 | 992 | 956 | 31.18 | 30.73 | 25.02 | 26.13 | 3.60 | 1.16 | 1.31 | 0.26 |
| | 0.04 | 898 | 903 | 990 | 965 | 31.37 | 30.80 | 25.02 | 26.18 | 4.38 | 1.45 | 1.29 | 0.26 |
| | 0.05 | 919 | 919 | 988 | 969 | 31.49 | 30.88 | 25.02 | 26.19 | 4.99 | 1.76 | 1.24 | 0.28 |
| | 0.06 | 943 | 933 | 986 | 975 | 31.60 | 30.95 | 25.02 | 26.22 | 5.48 | 2.05 | 1.21 | 0.28 |
| | 0.08 | 967 | 961 | 985 | 977 | 31.71 | 31.08 | 25.02 | 26.22 | 6.22 | 2.49 | 1.18 | 0.26 |
| | 0.10 | 981 | 968 | 981 | 979 | 31.80 | 31.13 | 25.00 | 26.23 | 6.39 | 2.83 | 1.22 | 0.25 |
| **MVCQT** | 0.12 | 993 | 979 | 981 | 980 | 31.80 | 31.20 | 25.01 | 26.23 | 6.22 | 3.23 | 1.17 | 0.24 |
| | 0.14 | 995 | 983 | 981 | 980 | 31.80 | 31.23 | 25.01 | 26.23 | 6.24 | 3.53 | 1.15 | 0.25 |
| | 0.15 | 994 | 994 | 980 | 981 | 31.81 | 31.29 | 25.00 | 26.23 | 6.37 | 3.68 | 1.16 | 0.25 |
| | 0.16 | 997 | 998 | 980 | 983 | 31.82 | 31.32 | 25.00 | 26.24 | 6.42 | 3.86 | 1.16 | 0.24 |
| | 0.18 | 996 | 1000 | 979 | 982 | 31.84 | 31.33 | 25.00 | 26.23 | 6.68 | 4.35 | 1.14 | 0.25 |
| | 0.20 | 997 | 1005 | 978 | 982 | 31.84 | 31.36 | 24.99 | 26.23 | 6.50 | 4.64 | 1.15 | 0.25 |
| | 0.40 | 997 | 1017 | 977 | 984 | 31.78 | 31.46 | 24.98 | 26.24 | 6.37 | 6.07 | 1.16 | 0.25 |
| | 0.60 | 1000 | 1018 | 981 | 985 | 31.80 | 31.47 | 24.98 | 26.24 | 6.71 | 6.15 | 1.12 | 0.28 |
| | 0.80 | 1016 | 1018 | 982 | 987 | 31.71 | 31.39 | 24.99 | 26.24 | 6.46 | 5.57 | 1.10 | 0.32 |
| | 0.999 | 1026 | 1016 | 981 | 985 | 31.61 | 31.34 | 24.98 | 26.23 | 5.59 | 5.49 | 1.13 | 0.37 |

Table 6.2: Simulation Results for IPPP GOV Structure @ 2000 kbps
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Q2** | – | 2006 | 2056 | 2002 | 2002 | 36.25 | 35.22 | 29.64 | 30.59 | 4.81 | 6.03 | 1.07 | 0.76 |
| **TM5** | – | 2001 | 2000 | 2001 | 2001 | 35.22 | 34.73 | 28.76 | 29.99 | 6.35 | 5.48 | 1.27 | 0.75 |
| **Q2T** | – | 2010 | 2057 | 2001 | 2003 | 35.14 | 34.61 | 28.70 | 29.72 | 3.52 | 6.18 | 1.22 | 0.57 |
| **TM5T** | – | 2001 | 2000 | 2001 | 2001 | 34.33 | 34.19 | 27.91 | 29.16 | 6.15 | 5.80 | 1.31 | 0.70 |
| | 0.00 | 1566 | 1581 | 1922 | 1906 | 33.94 | 33.22 | 28.28 | 29.41 | 0.48 | 0.59 | 1.26 | 0.55 |
| | 0.01 | 1648 | 1511 | 1920 | 1908 | 34.30 | 32.83 | 28.28 | 29.42 | 1.13 | 0.60 | 1.24 | 0.55 |
| | 0.02 | 1651 | 1560 | 1921 | 1909 | 34.31 | 32.99 | 28.28 | 29.42 | 1.40 | 0.65 | 1.24 | 0.55 |
| | 0.03 | 1664 | 1571 | 1921 | 1916 | 34.37 | 33.08 | 28.28 | 29.44 | 3.14 | 0.81 | 1.24 | 0.58 |
| | 0.04 | 1707 | 1586 | 1921 | 1930 | 34.55 | 33.11 | 28.28 | 29.48 | 3.17 | 0.97 | 1.24 | 0.67 |
| | 0.05 | 1736 | 1585 | 1921 | 1934 | 34.56 | 33.16 | 28.28 | 29.49 | 3.28 | 1.15 | 1.24 | 0.69 |
| | 0.06 | 1820 | 1634 | 1926 | 1933 | 34.73 | 33.34 | 28.30 | 29.49 | 4.20 | 1.48 | 1.23 | 0.68 |
| | 0.08 | 1881 | 1803 | 1920 | 1948 | 34.86 | 33.78 | 28.28 | 29.53 | 4.59 | 2.80 | 1.23 | 0.80 |
| | 0.10 | 1906 | 1819 | 1928 | 1946 | 34.88 | 33.82 | 28.30 | 29.52 | 5.32 | 3.27 | 1.24 | 0.79 |
| **FLCQT** | 0.12 | 1920 | 1853 | 1927 | 1958 | 34.91 | 33.92 | 28.30 | 29.54 | 5.55 | 3.70 | 1.25 | 0.83 |
| | 0.14 | 1933 | 1872 | 1926 | 1959 | 34.92 | 33.98 | 28.30 | 29.54 | 5.58 | 4.07 | 1.25 | 0.86 |
| | 0.15 | 1954 | 1894 | 1931 | 1959 | 34.94 | 34.04 | 28.31 | 29.55 | 5.36 | 4.28 | 1.24 | 0.87 |
| | 0.16 | 1964 | 1918 | 1931 | 1962 | 34.92 | 34.12 | 28.31 | 29.55 | 5.11 | 4.84 | 1.24 | 0.88 |
| | 0.18 | 1974 | 1925 | 1931 | 1963 | 34.99 | 34.14 | 28.31 | 29.55 | 5.42 | 5.23 | 1.24 | 0.89 |
| | 0.20 | 1983 | 1940 | 1944 | 1963 | 34.94 | 34.24 | 28.34 | 29.55 | 5.40 | 5.63 | 1.26 | 0.90 |
| | 0.40 | 1998 | 2003 | 1937 | 1971 | 34.95 | 34.36 | 28.32 | 29.57 | 5.21 | 6.66 | 1.28 | 0.95 |
| | 0.60 | 2001 | 2007 | 1939 | 1967 | 34.99 | 34.37 | 28.33 | 29.55 | 5.53 | 6.65 | 1.26 | 0.93 |
| | 0.80 | 1990 | 2008 | 1959 | 1969 | 34.96 | 34.34 | 28.34 | 29.55 | 5.49 | 6.62 | 1.32 | 0.96 |
| | 0.999 | 1996 | 2014 | 1959 | 1969 | 35.07 | 34.34 | 28.34 | 29.55 | 6.17 | 6.38 | 1.32 | 0.98 |
| | 0.00 | 1677 | 1587 | 2027 | 1860 | 34.10 | 32.98 | 28.58 | 29.24 | 0.33 | 0.34 | 1.64 | 0.30 |
| | 0.01 | 1763 | 1609 | 2008 | 1890 | 34.29 | 33.10 | 28.53 | 29.34 | 0.92 | 0.27 | 1.78 | 0.31 |
| | 0.02 | 1782 | 1641 | 2013 | 1917 | 34.38 | 33.22 | 28.56 | 29.43 | 1.98 | 0.25 | 1.53 | 0.38 |
| | 0.03 | 1813 | 1670 | 2004 | 1930 | 34.51 | 33.33 | 28.56 | 29.46 | 2.58 | 0.40 | 1.42 | 0.41 |
| | 0.04 | 1839 | 1708 | 2004 | 1937 | 34.56 | 33.46 | 28.59 | 29.48 | 3.78 | 0.61 | 1.23 | 0.44 |
| | 0.05 | 1878 | 1741 | 1999 | 1944 | 34.71 | 33.55 | 28.60 | 29.51 | 4.04 | 0.88 | 1.16 | 0.44 |
| | 0.06 | 1903 | 1775 | 1995 | 1950 | 34.76 | 33.64 | 28.59 | 29.53 | 4.31 | 1.17 | 1.11 | 0.47 |
| | 0.08 | 1945 | 1867 | 1988 | 1958 | 34.82 | 33.89 | 28.58 | 29.55 | 4.35 | 1.88 | 1.11 | 0.49 |
| | 0.10 | 1965 | 1928 | 1985 | 1962 | 34.84 | 34.06 | 28.56 | 29.57 | 4.58 | 2.37 | 1.10 | 0.52 |
| **MVCQT** | 0.12 | 1988 | 1948 | 1982 | 1964 | 34.90 | 34.12 | 28.55 | 29.57 | 4.51 | 2.81 | 1.10 | 0.52 |
| | 0.14 | 1996 | 1965 | 1979 | 1965 | 34.94 | 34.17 | 28.55 | 29.58 | 4.60 | 3.24 | 1.10 | 0.53 |
| | 0.15 | 2002 | 1974 | 1981 | 1966 | 34.96 | 34.21 | 28.55 | 29.58 | 4.46 | 3.47 | 1.08 | 0.53 |
| | 0.16 | 2004 | 1987 | 1979 | 1967 | 34.95 | 34.25 | 28.55 | 29.58 | 4.34 | 3.63 | 1.07 | 0.54 |
| | 0.18 | 1997 | 2001 | 1980 | 1967 | 34.93 | 34.26 | 28.56 | 29.59 | 4.59 | 3.95 | 1.05 | 0.53 |
| | 0.20 | 2003 | 2005 | 1979 | 1968 | 34.92 | 34.26 | 28.55 | 29.59 | 4.35 | 4.18 | 1.07 | 0.54 |
| | 0.40 | 2004 | 2039 | 1977 | 1970 | 34.80 | 34.39 | 28.54 | 29.58 | 4.52 | 5.35 | 1.07 | 0.59 |
| | 0.60 | 2001 | 2035 | 1981 | 1970 | 34.60 | 34.34 | 28.54 | 29.59 | 4.13 | 4.86 | 1.08 | 0.60 |
| | 0.80 | 1992 | 2033 | 1983 | 1987 | 34.69 | 34.36 | 28.54 | 29.56 | 4.19 | 5.42 | 1.09 | 0.56 |
| | 0.999 | 2008 | 2038 | 1981 | 1990 | 34.54 | 34.27 | 28.54 | 29.55 | 4.54 | 5.22 | 1.10 | 0.55 |

Table 6.3: Simulation Results for IPPP GOV Structure @ 3000 kbps
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Q2** | – | 3105 | 2999 | 3004 | 2999 | 39.06 | 37.24 | 32.34 | 33.00 | 0.96 | 4.45 | 1.54 | 0.82 |
| **TM5** | – | 3001 | 3000 | 3001 | 3001 | 37.46 | 36.63 | 31.32 | 32.28 | 4.71 | 4.78 | 1.44 | 0.85 |
| **Q2T** | – | 2887 | 2898 | 3002 | 3006 | 36.85 | 36.06 | 30.82 | 31.67 | 1.08 | 4.24 | 1.26 | 0.78 |
| **TM5T** | – | 3001 | 3000 | 3001 | 3001 | 36.29 | 35.89 | 30.17 | 31.27 | 4.40 | 5.10 | 1.63 | 0.94 |
| | 0.00 | 1919 | 2272 | 2847 | 2107 | 34.92 | 34.50 | 30.33 | 29.93 | 1.97 | 0.64 | 1.03 | 0.50 |
| | 0.01 | 2013 | 2299 | 2929 | 2153 | 35.06 | 34.61 | 30.37 | 30.05 | 2.23 | 0.59 | 0.92 | 0.66 |
| | 0.02 | 2416 | 2326 | 2917 | 2202 | 35.81 | 34.72 | 30.38 | 30.19 | 2.73 | 0.65 | 1.46 | 0.85 |
| | 0.03 | 2428 | 2382 | 2911 | 2218 | 35.83 | 34.86 | 30.35 | 30.25 | 2.67 | 0.73 | 1.51 | 0.89 |
| | 0.04 | 2540 | 2455 | 2910 | 2258 | 36.02 | 35.05 | 30.36 | 30.34 | 3.26 | 0.98 | 1.52 | 0.97 |
| | 0.05 | 2595 | 2585 | 2896 | 2859 | 36.10 | 35.30 | 30.32 | 31.33 | 3.50 | 1.58 | 1.59 | 0.88 |
| | 0.06 | 2619 | 2617 | 2882 | 2857 | 36.13 | 35.39 | 30.32 | 31.34 | 3.59 | 1.81 | 1.78 | 0.82 |
| | 0.08 | 2780 | 2698 | 2891 | 2865 | 36.40 | 35.54 | 30.30 | 31.35 | 3.91 | 2.38 | 1.77 | 0.88 |
| | 0.10 | 2892 | 2797 | 2892 | 2934 | 36.54 | 35.72 | 30.33 | 31.44 | 4.25 | 3.18 | 1.76 | 1.09 |
| **FLCQT** | 0.12 | 2959 | 2840 | 2886 | 2939 | 36.58 | 35.82 | 30.33 | 31.45 | 4.32 | 3.68 | 1.68 | 1.09 |
| | 0.14 | 2996 | 2839 | 2886 | 2951 | 36.62 | 35.88 | 30.33 | 31.42 | 4.20 | 3.94 | 1.68 | 1.26 |
| | 0.15 | 2933 | 2857 | 2886 | 2951 | 36.63 | 35.92 | 30.33 | 31.43 | 4.12 | 4.17 | 1.68 | 1.31 |
| | 0.16 | 2935 | 2910 | 2896 | 2951 | 36.61 | 35.97 | 30.34 | 31.43 | 4.17 | 4.43 | 1.69 | 1.28 |
| | 0.18 | 2936 | 2920 | 2896 | 2952 | 36.61 | 36.00 | 30.34 | 31.42 | 4.18 | 4.59 | 1.69 | 1.34 |
| | 0.20 | 2938 | 2924 | 2896 | 2952 | 36.63 | 36.04 | 30.34 | 31.43 | 4.28 | 4.79 | 1.69 | 1.31 |
| | 0.40 | 2937 | 3014 | 2905 | 2955 | 36.59 | 36.13 | 30.34 | 31.43 | 4.22 | 5.52 | 1.77 | 1.32 |
| | 0.60 | 2939 | 3017 | 2906 | 2956 | 36.65 | 36.13 | 30.31 | 31.44 | 4.42 | 5.48 | 1.68 | 1.34 |
| | 0.80 | 2907 | 3019 | 2905 | 2957 | 36.64 | 36.11 | 30.30 | 31.43 | 4.53 | 5.41 | 1.81 | 1.42 |
| | 0.999 | 2903 | 3014 | 2914 | 2958 | 36.65 | 36.10 | 30.30 | 31.45 | 4.36 | 5.39 | 1.82 | 1.35 |
| | 0.00 | 2381 | 2354 | 2931 | 2751 | 35.36 | 34.50 | 30.12 | 31.22 | 0.93 | 0.23 | 2.28 | 0.17 |
| | 0.01 | 2521 | 2417 | 2926 | 2825 | 35.65 | 34.69 | 30.11 | 31.35 | 1.26 | 0.10 | 2.37 | 0.25 |
| | 0.02 | 2634 | 2472 | 2928 | 2886 | 35.93 | 34.86 | 30.15 | 31.47 | 2.18 | 0.17 | 2.10 | 0.31 |
| | 0.03 | 2704 | 2549 | 2916 | 2913 | 36.07 | 35.04 | 30.14 | 31.52 | 2.74 | 0.40 | 2.13 | 0.37 |
| | 0.04 | 2751 | 2631 | 2927 | 2918 | 36.16 | 35.20 | 30.21 | 31.52 | 3.26 | 0.79 | 1.82 | 0.37 |
| | 0.05 | 2844 | 2679 | 2930 | 2925 | 36.25 | 35.30 | 30.22 | 31.53 | 3.44 | 1.07 | 1.74 | 0.39 |
| | 0.06 | 2918 | 2762 | 2921 | 2933 | 36.33 | 35.40 | 30.20 | 31.54 | 3.72 | 1.40 | 1.79 | 0.38 |
| | 0.08 | 2979 | 2867 | 2930 | 2940 | 36.44 | 35.52 | 30.27 | 31.55 | 3.73 | 2.00 | 1.37 | 0.39 |
| | 0.10 | 3008 | 2918 | 2952 | 2944 | 36.48 | 35.59 | 30.38 | 31.55 | 3.54 | 2.49 | 0.90 | 0.35 |
| **MVCQT** | 0.12 | 3021 | 2930 | 2945 | 2946 | 36.41 | 35.59 | 30.37 | 31.55 | 4.04 | 2.94 | 0.94 | 0.31 |
| | 0.14 | 3042 | 2967 | 2941 | 2947 | 36.45 | 35.71 | 30.36 | 31.55 | 3.64 | 3.09 | 0.96 | 0.33 |
| | 0.15 | 3036 | 2974 | 2938 | 2949 | 36.47 | 35.70 | 30.35 | 31.56 | 3.58 | 3.48 | 0.99 | 0.30 |
| | 0.16 | 3037 | 2996 | 2939 | 2948 | 36.39 | 35.76 | 30.36 | 31.55 | 3.94 | 3.45 | 0.95 | 0.31 |
| | 0.18 | 3035 | 3015 | 2937 | 2951 | 36.42 | 35.76 | 30.36 | 31.56 | 3.71 | 3.66 | 0.93 | 0.30 |
| | 0.20 | 3034 | 3009 | 2934 | 2951 | 36.45 | 35.73 | 30.36 | 31.55 | 3.36 | 3.90 | 0.96 | 0.30 |
| | 0.40 | 3038 | 3054 | 2924 | 2955 | 36.37 | 35.86 | 30.35 | 31.56 | 3.58 | 4.40 | 1.01 | 0.30 |
| | 0.60 | 3034 | 3050 | 2916 | 2962 | 36.35 | 35.85 | 30.32 | 31.57 | 3.77 | 4.60 | 1.13 | 0.34 |
| | 0.80 | 3033 | 3050 | 2913 | 2955 | 36.31 | 35.70 | 30.31 | 31.53 | 3.67 | 3.75 | 1.20 | 0.46 |
| | 0.999 | 3044 | 3045 | 2908 | 2956 | 36.27 | 35.72 | 30.29 | 31.53 | 4.07 | 4.20 | 1.29 | 0.54 |

Table 6.4: Simulation Results for IBBP GOV Structure @ 1000 kbps
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Q2** | – | 1002 | 1014 | 1005 | 1004 | 31.74 | 31.28 | 24.68 | 26.32 | 3.99 | 7.48 | 1.60 | 0.92 |
| **TM5** | – | 1000 | 1000 | 1001 | 1000 | 32.39 | 31.63 | 26.14 | 28.42 | 5.62 | 8.31 | 0.91 | 0.95 |
| **Q2T** | – | 1003 | 1011 | 1007 | 1002 | 31.33 | 30.89 | 24.47 | 26.12 | 3.50 | 7.41 | 1.46 | 0.72 |
| **TM5T** | – | 1000 | 1000 | 1001 | 1000 | 32.12 | 31.44 | 25.92 | 28.19 | 5.33 | 8.00 | 0.92 | 0.96 |
| | 0.00 | 767 | 842 | 970 | 706 | 31.12 | 30.23 | 25.95 | 26.13 | 0.49 | 1.20 | 0.36 | 0.07 |
| | 0.01 | 812 | 851 | 970 | 875 | 31.40 | 30.28 | 25.95 | 27.05 | 1.09 | 1.29 | 0.36 | 0.42 |
| | 0.02 | 862 | 864 | 977 | 938 | 31.61 | 30.36 | 26.08 | 27.39 | 1.82 | 1.55 | 0.29 | 0.39 |
| | 0.03 | 900 | 875 | 978 | 944 | 31.79 | 30.44 | 26.10 | 27.43 | 2.94 | 1.73 | 0.29 | 0.42 |
| | 0.04 | 925 | 882 | 977 | 957 | 31.91 | 30.49 | 26.08 | 27.47 | 2.94 | 2.08 | 0.31 | 0.45 |
| | 0.05 | 941 | 896 | 979 | 970 | 32.08 | 30.53 | 26.09 | 27.52 | 3.76 | 2.31 | 0.30 | 0.47 |
| | 0.06 | 951 | 909 | 975 | 974 | 32.09 | 30.60 | 26.03 | 27.53 | 4.05 | 2.63 | 0.34 | 0.47 |
| | 0.08 | 982 | 933 | 975 | 980 | 32.25 | 30.64 | 26.03 | 27.49 | 4.47 | 3.19 | 0.34 | 0.68 |
| | 0.10 | 985 | 937 | 975 | 982 | 32.32 | 30.71 | 26.03 | 27.50 | 4.65 | 3.63 | 0.34 | 0.67 |
| **FLCQT** | 0.12 | 997 | 948 | 969 | 981 | 32.41 | 30.76 | 25.98 | 27.53 | 4.81 | 3.97 | 0.42 | 0.70 |
| | 0.14 | 997 | 963 | 979 | 982 | 32.34 | 30.82 | 26.08 | 27.49 | 4.65 | 4.47 | 0.37 | 0.67 |
| | 0.15 | 999 | 968 | 979 | 981 | 32.35 | 30.87 | 26.09 | 27.49 | 4.83 | 4.62 | 0.36 | 0.61 |
| | 0.16 | 1000 | 971 | 979 | 981 | 32.30 | 30.87 | 26.09 | 27.52 | 4.42 | 4.60 | 0.36 | 0.65 |
| | 0.18 | 1001 | 978 | 979 | 981 | 32.33 | 30.93 | 26.07 | 27.53 | 4.34 | 5.13 | 0.39 | 0.64 |
| | 0.20 | 1002 | 989 | 979 | 982 | 32.33 | 31.00 | 26.07 | 27.59 | 4.31 | 5.92 | 0.39 | 0.77 |
| | 0.40 | 1000 | 1014 | 974 | 985 | 32.40 | 31.19 | 25.95 | 27.57 | 4.75 | 7.35 | 0.45 | 0.83 |
| | 0.60 | 1000 | 1016 | 966 | 985 | 32.28 | 31.21 | 25.94 | 27.71 | 4.64 | 7.88 | 0.44 | 0.89 |
| | 0.80 | 1001 | 1018 | 964 | 987 | 32.28 | 31.20 | 25.97 | 27.63 | 4.32 | 8.02 | 0.41 | 0.78 |
| | 0.999 | 1002 | 1015 | 967 | 986 | 32.30 | 31.17 | 25.96 | 27.60 | 4.90 | 7.66 | 0.56 | 0.88 |
| | 0.00 | 782 | 761 | 878 | 732 | 30.34 | 29.40 | 24.84 | 26.05 | 0.41 | 1.74 | 1.28 | 1.03 |
| | 0.01 | 873 | 774 | 889 | 808 | 30.82 | 29.56 | 24.90 | 26.45 | 0.93 | 1.87 | 1.22 | 0.81 |
| | 0.02 | 899 | 801 | 904 | 921 | 30.97 | 29.75 | 25.01 | 27.13 | 1.83 | 2.30 | 1.14 | 0.46 |
| | 0.03 | 925 | 837 | 917 | 983 | 31.14 | 30.00 | 25.09 | 27.48 | 2.44 | 2.68 | 1.22 | 0.37 |
| | 0.04 | 974 | 865 | 923 | 987 | 31.40 | 30.19 | 25.10 | 27.50 | 3.15 | 2.85 | 1.28 | 0.36 |
| | 0.05 | 980 | 889 | 936 | 988 | 31.45 | 30.36 | 25.13 | 27.51 | 3.52 | 3.15 | 1.07 | 0.37 |
| | 0.06 | 988 | 916 | 954 | 989 | 31.43 | 30.49 | 25.32 | 27.55 | 3.50 | 3.69 | 0.89 | 0.33 |
| | 0.08 | 997 | 951 | 992 | 990 | 31.50 | 30.66 | 25.78 | 27.55 | 3.34 | 4.48 | 0.41 | 0.34 |
| | 0.10 | 1006 | 970 | 985 | 990 | 31.56 | 30.80 | 25.78 | 27.55 | 3.05 | 5.36 | 0.50 | 0.34 |
| **MVCQT** | 0.12 | 1011 | 986 | 984 | 989 | 31.64 | 30.88 | 25.84 | 27.55 | 3.24 | 5.85 | 0.50 | 0.34 |
| | 0.14 | 1009 | 993 | 983 | 989 | 31.53 | 30.94 | 25.84 | 27.55 | 3.09 | 6.28 | 0.50 | 0.34 |
| | 0.15 | 1014 | 997 | 983 | 990 | 31.61 | 30.96 | 25.84 | 27.55 | 3.29 | 6.40 | 0.51 | 0.35 |
| | 0.16 | 1013 | 1001 | 982 | 990 | 31.66 | 30.98 | 25.84 | 27.56 | 3.22 | 6.53 | 0.53 | 0.35 |
| | 0.18 | 1015 | 1003 | 982 | 990 | 31.65 | 31.02 | 25.84 | 27.57 | 3.14 | 6.84 | 0.51 | 0.36 |
| | 0.20 | 1013 | 1003 | 982 | 990 | 31.67 | 31.03 | 25.84 | 27.56 | 3.26 | 6.88 | 0.51 | 0.35 |
| | 0.40 | 1009 | 1018 | 983 | 990 | 31.64 | 31.09 | 25.86 | 27.57 | 3.45 | 8.28 | 0.55 | 0.32 |
| | 0.60 | 1002 | 1016 | 983 | 993 | 31.55 | 31.06 | 25.87 | 27.55 | 3.37 | 8.18 | 0.64 | 0.45 |
| | 0.80 | 1000 | 1016 | 982 | 996 | 31.55 | 31.04 | 25.86 | 27.56 | 3.72 | 7.66 | 0.64 | 0.57 |
| | 0.999 | 1001 | 1015 | 981 | 995 | 31.47 | 30.89 | 25.82 | 27.56 | 3.49 | 6.79 | 0.66 | 0.33 |

Table 6.5: Simulation Results for IBBP GOV Structure @ 2000 kbps
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Q2** | – | 2005 | 2036 | 2008 | 2006 | 35.07 | 34.64 | 28.70 | 30.56 | 3.52 | 7.62 | 1.96 | 0.75 |
| **TM5** | – | 2001 | 2000 | 2001 | 2000 | 35.76 | 34.61 | 29.62 | 31.53 | 4.69 | 7.51 | 1.84 | 1.74 |
| **Q2T** | – | 2004 | 2039 | 2012 | 2003 | 34.46 | 33.84 | 28.06 | 29.87 | 2.69 | 6.91 | 1.34 | 0.55 |
| **TM5T** | – | 2000 | 2000 | 2001 | 2000 | 35.19 | 34.15 | 29.03 | 30.98 | 5.46 | 7.32 | 1.62 | 1.48 |
| | 0.00 | 1654 | 1178 | 1940 | 1582 | 34.43 | 31.44 | 29.06 | 29.74 | 0.88 | 0.81 | 0.83 | 0.27 |
| | 0.01 | 1702 | 1370 | 1933 | 1638 | 34.64 | 32.30 | 29.10 | 29.92 | 1.35 | 1.34 | 0.78 | 0.33 |
| | 0.02 | 1740 | 1550 | 1944 | 1717 | 34.73 | 32.88 | 29.26 | 30.06 | 1.56 | 1.96 | 0.86 | 0.41 |
| | 0.03 | 1778 | 1537 | 1925 | 1848 | 34.93 | 32.81 | 29.24 | 30.51 | 2.14 | 1.93 | 0.85 | 0.56 |
| | 0.04 | 1822 | 1562 | 1913 | 1913 | 35.08 | 32.97 | 29.23 | 30.75 | 2.71 | 2.25 | 0.83 | 0.91 |
| | 0.05 | 1928 | 1555 | 1946 | 1921 | 35.23 | 32.99 | 29.29 | 30.79 | 3.63 | 2.28 | 0.93 | 0.98 |
| | 0.06 | 1953 | 1644 | 1963 | 1944 | 35.27 | 33.21 | 29.34 | 30.82 | 3.55 | 2.85 | 0.99 | 0.85 |
| | 0.08 | 1965 | 1771 | 1963 | 1954 | 35.34 | 33.48 | 29.37 | 30.82 | 3.84 | 3.38 | 0.93 | 0.86 |
| | 0.10 | 1977 | 1851 | 1956 | 1958 | 35.28 | 33.65 | 29.28 | 30.79 | 3.71 | 3.88 | 0.92 | 0.93 |
| **FLCQT** | 0.12 | 1984 | 1908 | 1954 | 1967 | 35.32 | 33.78 | 29.27 | 30.97 | 3.73 | 4.83 | 0.90 | 1.10 |
| | 0.14 | 1986 | 1916 | 1958 | 1960 | 35.32 | 33.82 | 29.32 | 30.84 | 3.79 | 5.00 | 1.10 | 1.08 |
| | 0.15 | 1987 | 1930 | 1960 | 1967 | 35.32 | 33.91 | 29.41 | 30.94 | 3.79 | 5.15 | 1.19 | 1.00 |
| | 0.16 | 1988 | 1933 | 1962 | 1963 | 35.27 | 33.94 | 29.33 | 30.78 | 4.02 | 5.53 | 1.01 | 0.92 |
| | 0.18 | 1988 | 1994 | 1957 | 1964 | 35.46 | 34.01 | 29.27 | 30.81 | 4.32 | 6.07 | 0.85 | 0.95 |
| | 0.20 | 1999 | 1994 | 1960 | 1966 | 35.35 | 34.02 | 29.39 | 30.85 | 3.75 | 6.06 | 1.07 | 0.96 |
| | 0.40 | 1999 | 2043 | 1959 | 1972 | 35.50 | 34.28 | 29.38 | 30.90 | 4.37 | 7.60 | 1.23 | 0.98 |
| | 0.60 | 2002 | 2036 | 1954 | 1977 | 35.43 | 34.23 | 29.33 | 30.93 | 3.96 | 7.87 | 1.20 | 1.09 |
| | 0.80 | 2000 | 2034 | 1953 | 1977 | 35.26 | 34.14 | 29.30 | 30.96 | 3.67 | 7.56 | 1.05 | 1.51 |
| | 0.999 | 1999 | 2030 | 1960 | 1981 | 35.42 | 34.21 | 29.28 | 30.97 | 4.44 | 7.27 | 1.17 | 1.78 |
| | 0.00 | 1434 | 1153 | 1681 | 1706 | 32.80 | 31.02 | 27.59 | 29.79 | 0.47 | 0.59 | 1.96 | 0.18 |
| | 0.01 | 1605 | 1276 | 1916 | 1962 | 33.15 | 31.59 | 28.46 | 30.33 | 1.55 | 0.99 | 0.85 | 0.47 |
| | 0.02 | 1731 | 1414 | 1905 | 2014 | 33.54 | 32.08 | 28.44 | 30.36 | 2.43 | 1.82 | 0.89 | 0.50 |
| | 0.03 | 1864 | 1594 | 1996 | 2016 | 33.95 | 32.60 | 28.79 | 30.36 | 2.93 | 2.92 | 0.78 | 0.49 |
| | 0.04 | 1936 | 1678 | 1993 | 2020 | 34.04 | 32.76 | 28.80 | 30.38 | 3.99 | 3.19 | 0.76 | 0.50 |
| | 0.05 | 1968 | 1733 | 1993 | 2013 | 34.04 | 32.89 | 28.83 | 30.36 | 3.87 | 3.62 | 0.66 | 0.45 |
| | 0.06 | 1984 | 1767 | 1986 | 2015 | 34.07 | 32.89 | 28.85 | 30.35 | 3.63 | 3.49 | 0.61 | 0.46 |
| | 0.08 | 2007 | 1881 | 1981 | 2024 | 34.06 | 33.04 | 28.88 | 30.39 | 3.45 | 4.31 | 0.55 | 0.55 |
| | 0.10 | 2020 | 1910 | 1981 | 2024 | 34.16 | 33.19 | 28.87 | 30.38 | 3.51 | 5.11 | 0.53 | 0.53 |
| **MVCQT** | 0.12 | 2029 | 1944 | 1978 | 2027 | 34.18 | 33.28 | 28.87 | 30.38 | 3.57 | 5.62 | 0.53 | 0.60 |
| | 0.14 | 2031 | 1976 | 1976 | 2028 | 34.22 | 33.40 | 28.87 | 30.38 | 3.43 | 6.45 | 0.54 | 0.62 |
| | 0.15 | 2030 | 1984 | 1974 | 2027 | 34.20 | 33.42 | 28.87 | 30.37 | 3.57 | 6.72 | 0.55 | 0.57 |
| | 0.16 | 2031 | 1990 | 1973 | 2028 | 34.22 | 33.45 | 28.86 | 30.37 | 3.43 | 6.80 | 0.55 | 0.59 |
| | 0.18 | 2030 | 1992 | 1976 | 2027 | 34.22 | 33.45 | 28.88 | 30.37 | 3.40 | 7.13 | 0.51 | 0.63 |
| | 0.20 | 2028 | 2003 | 1969 | 2028 | 34.22 | 33.52 | 28.87 | 30.40 | 3.42 | 7.61 | 0.57 | 0.61 |
| | 0.40 | 2031 | 2037 | 1967 | 2026 | 34.02 | 33.53 | 28.87 | 30.36 | 4.62 | 7.94 | 0.60 | 0.64 |
| | 0.60 | 2022 | 2033 | 1964 | 2026 | 34.02 | 33.44 | 28.85 | 30.34 | 4.17 | 7.39 | 0.65 | 0.71 |
| | 0.80 | 2001 | 2033 | 1959 | 2007 | 33.82 | 33.28 | 28.82 | 30.30 | 3.79 | 7.29 | 0.72 | 0.77 |
| | 0.999 | 1990 | 2045 | 1951 | 2004 | 33.72 | 33.21 | 28.80 | 30.26 | 4.80 | 6.57 | 0.85 | 0.82 |

Table 6.6: Simulation Results for IBBP GOV Structure @ 3000 kbps
for table tennis (Seq. 1), football (Seq. 2), flower garden (Seq. 3), and mobile calendar (Seq.4)

| | | Actual Bitrate (kbps) | | | | Average PSNR (dB) | | | | PSNR Variance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\lambda$ | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 | Seq. 1 | Seq. 2 | Seq. 3 | Seq. 4 |
| **Q2** | – | 3130 | 2993 | 3017 | 3010 | 39.00 | 36.85 | 31.65 | 33.12 | 1.36 | 5.54 | 2.54 | 1.24 |
| **TM5** | – | 3000 | 3000 | 3001 | 3000 | 37.93 | 36.44 | 31.91 | 33.65 | 4.86 | 7.10 | 2.81 | 2.37 |
| **Q2T** | – | 2841 | 2900 | 3018 | 2999 | 36.67 | 35.22 | 30.06 | 31.80 | 0.98 | 5.43 | 1.41 | 0.57 |
| **TM5T** | – | 3001 | 3000 | 3002 | 3000 | 36.99 | 35.82 | 31.17 | 33.03 | 4.35 | 6.76 | 2.43 | 2.13 |
| | 0.00 | 2181 | 2242 | 1889 | 1644 | 35.43 | 34.64 | 29.00 | 29.92 | 0.84 | 1.41 | 0.89 | 0.52 |
| | 0.01 | 2232 | 2295 | 1998 | 1727 | 35.67 | 34.71 | 29.39 | 30.15 | 1.02 | 1.53 | 0.93 | 0.59 |
| | 0.02 | 2633 | 2555 | 2243 | 2673 | 36.57 | 35.18 | 29.86 | 32.45 | 2.44 | 2.73 | 1.14 | 1.89 |
| | 0.03 | 2695 | 2655 | 2280 | 2775 | 36.76 | 35.36 | 29.98 | 32.59 | 3.01 | 3.60 | 1.21 | 1.84 |
| | 0.04 | 2832 | 2721 | 2576 | 2790 | 36.90 | 35.43 | 30.48 | 32.62 | 3.14 | 4.14 | 1.46 | 1.72 |
| | 0.05 | 2886 | 2672 | 2709 | 2892 | 37.08 | 35.40 | 30.73 | 32.77 | 3.81 | 3.55 | 1.85 | 1.53 |
| | 0.06 | 2892 | 2721 | 2732 | 2886 | 37.13 | 35.46 | 30.80 | 32.79 | 3.80 | 3.88 | 2.01 | 1.66 |
| | 0.08 | 2897 | 2833 | 2798 | 2904 | 37.15 | 35.64 | 30.93 | 32.86 | 3.97 | 4.94 | 2.22 | 1.69 |
| | 0.10 | 2933 | 2897 | 2847 | 2916 | 37.28 | 35.75 | 31.07 | 32.89 | 4.28 | 5.33 | 2.57 | 1.65 |
| **FLCQT** | 0.12 | 2957 | 2928 | 2842 | 2931 | 37.25 | 35.83 | 31.07 | 32.89 | 4.30 | 5.62 | 2.62 | 1.77 |
| | 0.14 | 2934 | 2906 | 2862 | 2931 | 37.29 | 35.80 | 31.13 | 32.89 | 4.32 | 5.28 | 2.77 | 1.77 |
| | 0.15 | 2934 | 2891 | 2876 | 2931 | 37.29 | 35.81 | 31.15 | 32.89 | 4.32 | 5.36 | 2.75 | 1.77 |
| | 0.16 | 2978 | 2970 | 2862 | 2934 | 37.34 | 35.88 | 31.12 | 32.94 | 4.58 | 5.77 | 2.67 | 1.75 |
| | 0.18 | 2924 | 2942 | 2869 | 2933 | 37.32 | 35.86 | 31.13 | 32.93 | 4.43 | 5.61 | 2.65 | 1.76 |
| | 0.20 | 2989 | 2976 | 2874 | 2934 | 37.35 | 35.93 | 31.14 | 32.95 | 4.49 | 6.03 | 2.66 | 1.81 |
| | 0.40 | 2990 | 3018 | 2893 | 2934 | 37.34 | 35.98 | 31.17 | 32.89 | 4.65 | 6.48 | 2.79 | 1.78 |
| | 0.60 | 2992 | 3022 | 2921 | 2937 | 37.33 | 35.92 | 31.23 | 32.94 | 4.82 | 6.20 | 2.86 | 1.89 |
| | 0.80 | 2998 | 3026 | 2918 | 2938 | 37.33 | 35.98 | 31.23 | 32.94 | 4.82 | 6.47 | 2.86 | 1.94 |
| | 0.999 | 2996 | 3016 | 2958 | 2939 | 37.34 | 35.94 | 31.27 | 32.96 | 4.80 | 6.42 | 2.71 | 2.02 |
| | 0.00 | 2135 | 1695 | 2920 | 2207 | 34.05 | 32.47 | 29.97 | 30.79 | 1.08 | 0.45 | 2.22 | 0.24 |
| | 0.01 | 2449 | 1916 | 2930 | 2891 | 34.55 | 33.09 | 30.06 | 31.77 | 2.11 | 0.79 | 1.83 | 0.71 |
| | 0.02 | 2690 | 2140 | 2946 | 2925 | 34.95 | 33.60 | 30.16 | 31.81 | 3.58 | 1.52 | 0.65 | 0.66 |
| | 0.03 | 2839 | 2401 | 2911 | 2954 | 35.15 | 34.00 | 29.98 | 31.85 | 4.11 | 2.47 | 2.03 | 0.67 |
| | 0.04 | 2866 | 2495 | 2915 | 2973 | 35.18 | 34.02 | 30.10 | 31.85 | 4.13 | 2.48 | 1.43 | 0.68 |
| | 0.05 | 2929 | 2578 | 2922 | 2985 | 35.28 | 34.14 | 30.21 | 31.87 | 4.28 | 2.95 | 1.10 | 0.70 |
| | 0.06 | 2972 | 2705 | 2893 | 2994 | 35.35 | 34.28 | 30.10 | 31.87 | 4.47 | 3.74 | 1.37 | 0.71 |
| | 0.08 | 3021 | 2893 | 2960 | 2996 | 35.36 | 34.43 | 30.32 | 31.88 | 4.77 | 6.10 | 0.60 | 0.71 |
| | 0.10 | 3047 | 2960 | 2956 | 2997 | 35.36 | 34.54 | 30.35 | 31.88 | 4.43 | 6.94 | 0.62 | 0.73 |
| **MVCQT** | 0.12 | 3041 | 2985 | 2946 | 2998 | 35.32 | 34.65 | 30.31 | 31.88 | 4.42 | 6.59 | 0.62 | 0.73 |
| | 0.14 | 3046 | 3005 | 2884 | 2996 | 35.36 | 34.64 | 30.04 | 31.88 | 3.93 | 7.39 | 1.35 | 0.74 |
| | 0.15 | 3045 | 3000 | 2922 | 2995 | 35.36 | 34.68 | 30.21 | 31.89 | 4.07 | 6.74 | 0.79 | 0.75 |
| | 0.16 | 3046 | 3007 | 2890 | 2990 | 35.31 | 34.69 | 30.06 | 31.88 | 4.12 | 6.85 | 1.21 | 0.74 |
| | 0.18 | 3047 | 3030 | 2889 | 2991 | 35.45 | 34.63 | 30.06 | 31.88 | 3.83 | 7.74 | 1.24 | 0.77 |
| | 0.20 | 3048 | 3049 | 2914 | 3000 | 35.41 | 34.78 | 30.19 | 31.89 | 4.00 | 7.20 | 0.86 | 0.76 |
| | 0.40 | 3036 | 3059 | 2903 | 2990 | 35.38 | 34.72 | 30.11 | 31.85 | 4.06 | 6.62 | 1.07 | 0.81 |
| | 0.60 | 3035 | 3057 | 2911 | 3015 | 35.21 | 34.62 | 30.24 | 31.81 | 4.42 | 6.52 | 0.80 | 0.93 |
| | 0.80 | 3034 | 3051 | 2901 | 2994 | 35.15 | 34.35 | 30.07 | 31.78 | 4.43 | 6.54 | 1.34 | 1.01 |
| | 0.999 | 3032 | 3054 | 2899 | 2989 | 35.08 | 34.31 | 30.08 | 31.84 | 5.26 | 6.06 | 1.17 | 0.59 |

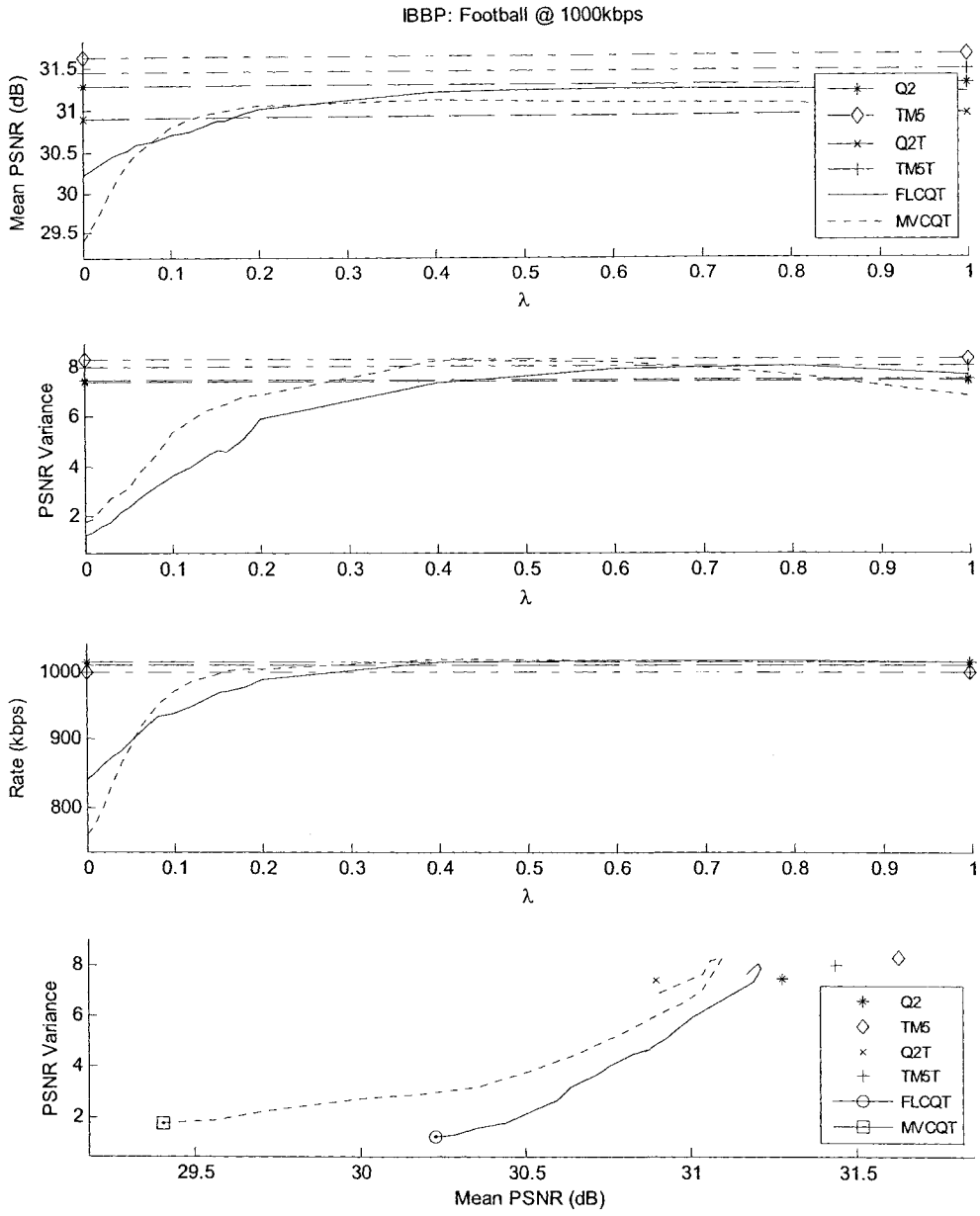



Figure 6.2: Effect of $\lambda$ for Football for Q2, TM5, Q2T, TM5T, FLCQT, and MVCQT algorithms (GOV: IBBP, Rate: 1000 kbps)

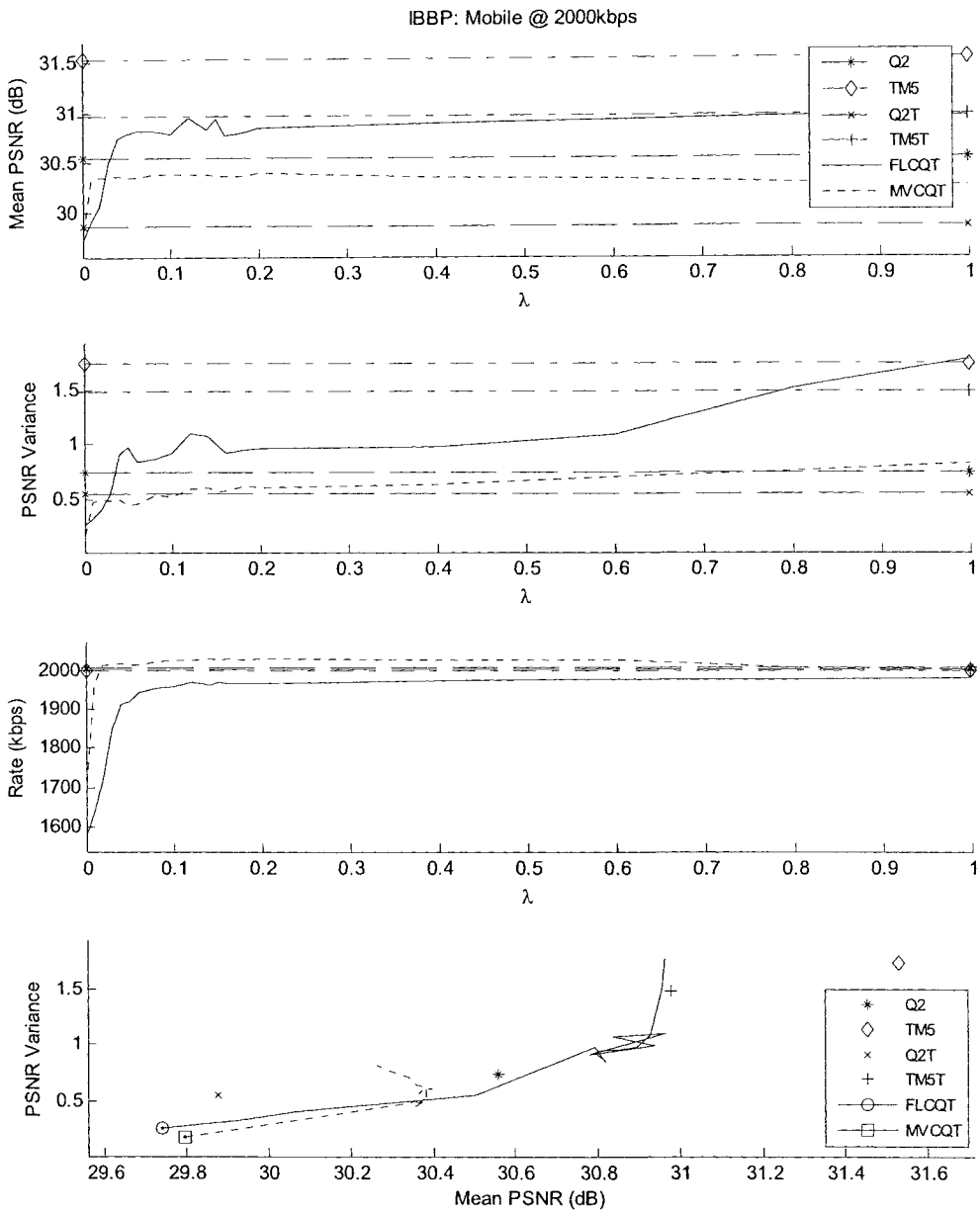



Figure 6.3: Effect of $\lambda$ for Mobile Calendar for Q2, TM5, Q2T, TM5T, FLCQT, and MVCQT algorithms (GOV: IBBP, Rate: 2000 kbps)

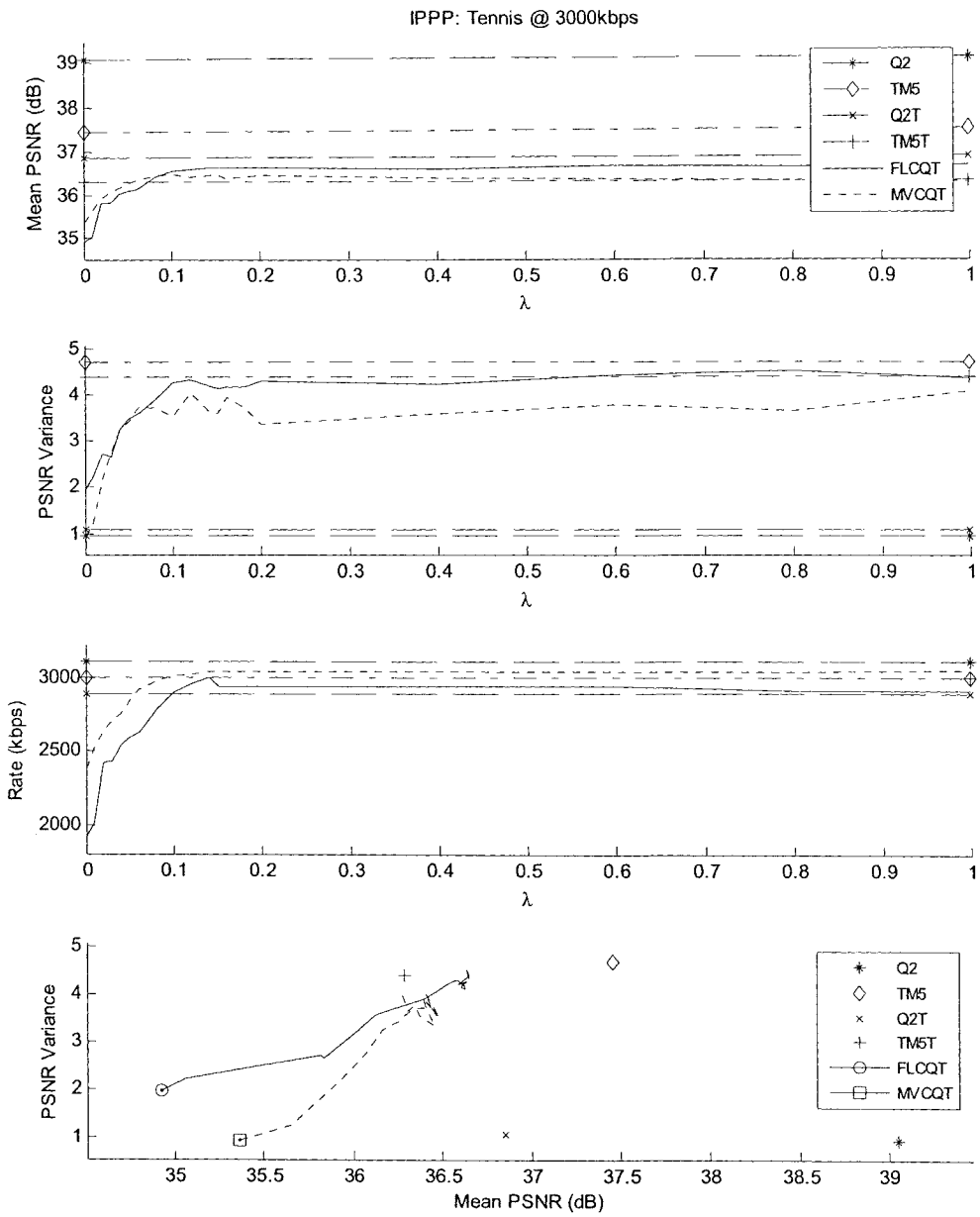



Figure 6.4: Effect of $\lambda$ for Table Tennis for Q2, TM5, Q2T, TM5T, FLCQT, and MVCQT algorithms (GOV: IPPP, Rate: 3000 kbps)

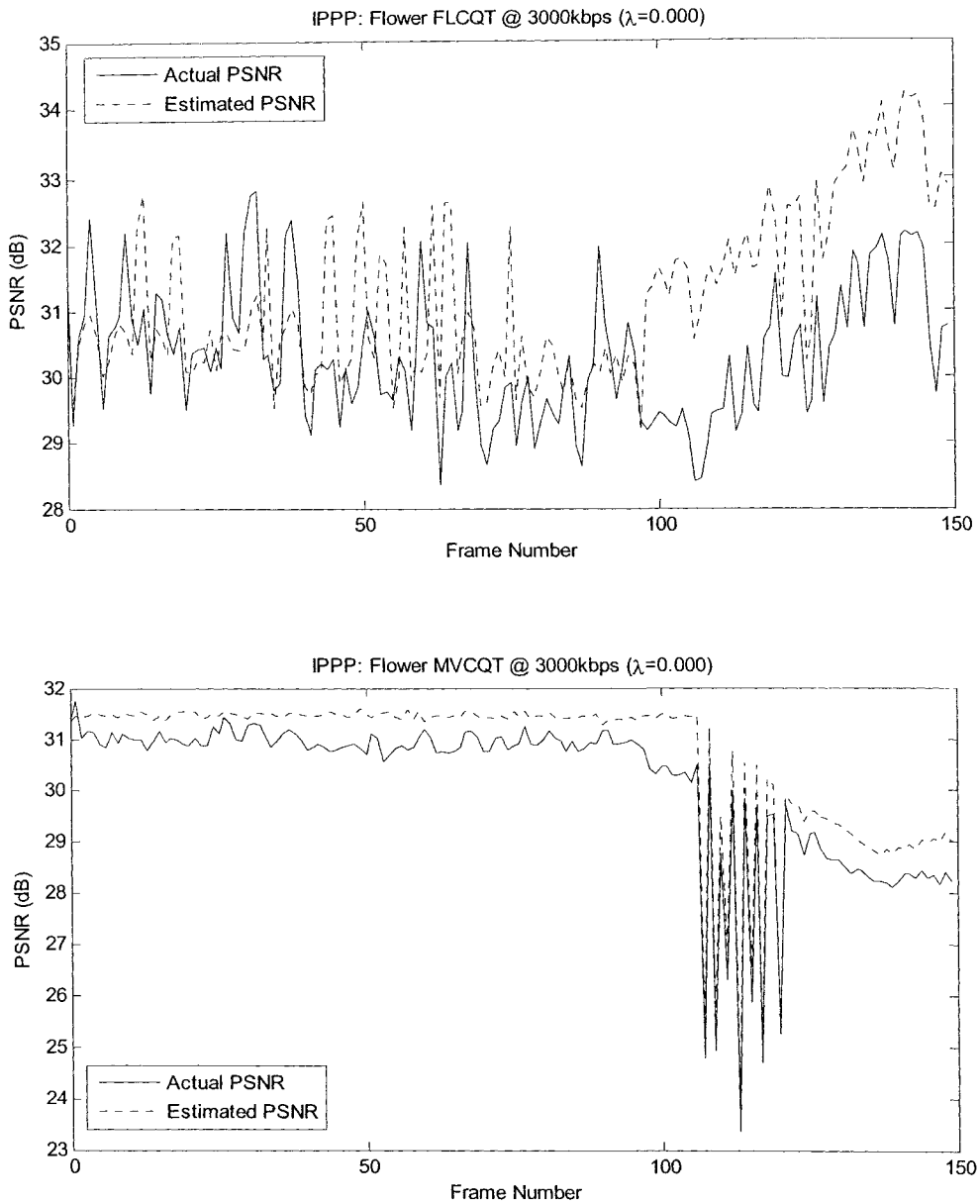



Figure 6.5: Actual and estimated PSNR by FLCQT and MVCQT algorithms for Flower Garden (GOV: IPPP, Rate=3000 kbps, $\lambda$=0)

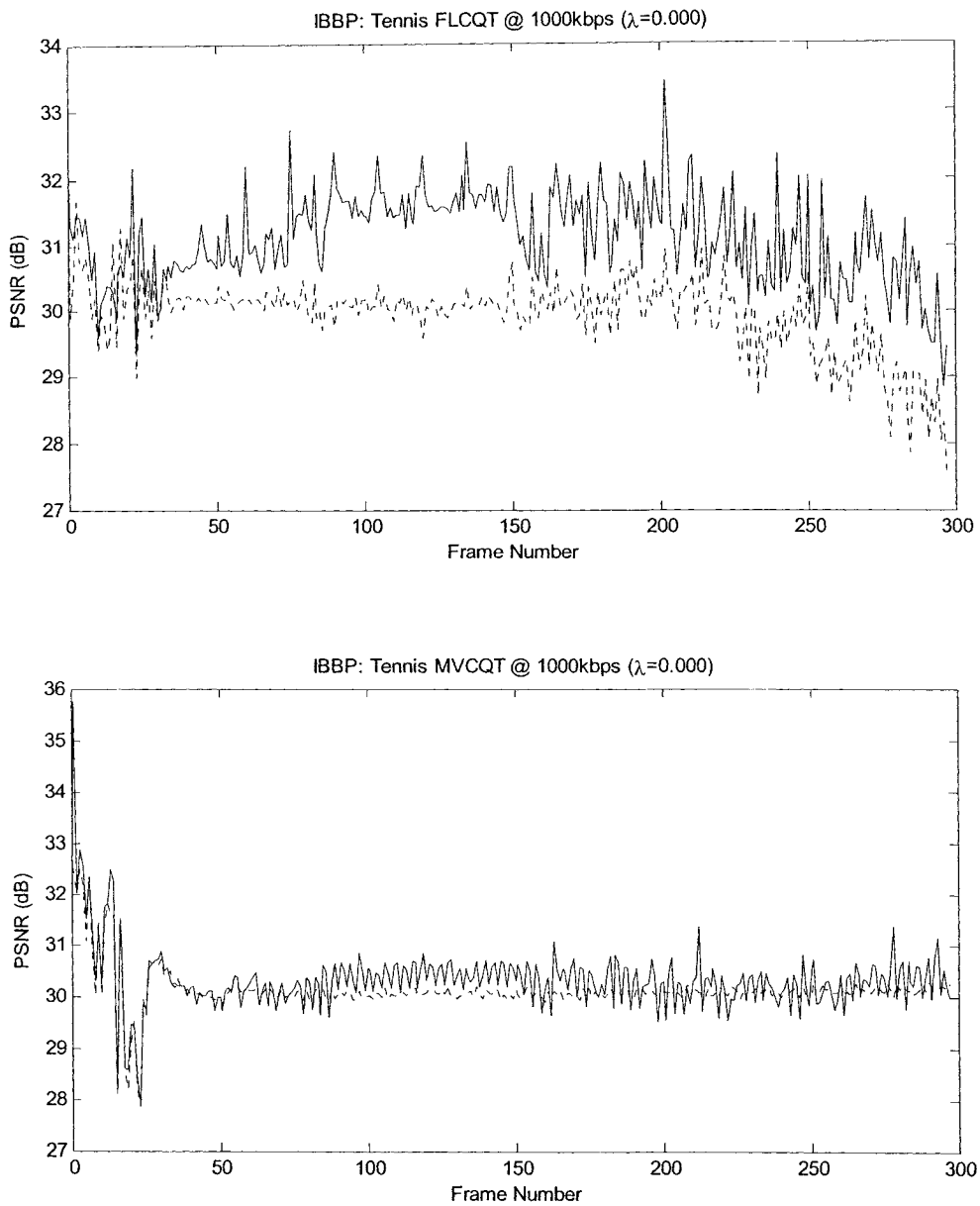



Figure 6.6: Actual and estimated PSNR by FLCQT and MVCQT algorithms for Table Tennis (GOV: IPPP, Rate=3000 kbps, $\lambda$=0)

## 6.4. Chapter Summary

In this chapter, two CQ CBR rate-control algorithms for MPEG-4 video transcoding are proposed. The FLCQT algorithm controls the quality at the frame level, and the MVCQT algorithm controls the quality at the macroblock level. These two algorithms are extended from the CQ CBR algorithms for MPEG-4 encoding proposed in Section 5.4. The main difference is that the transcoding algorithms use the QM/BM algorithms for transcoding while the encoding algorithms use the QM/BM algorithms for encoding.

As shown in the results, the proposed algorithms give smaller PSNR variance and similar (or slightly lower) average PSNR at a lower bitrate when $\lambda$ is small. As $\lambda$ increases, the actual bitrate meets the target bitrate more accurately, and both the average PSNR and the PSNR variance increase. This is expected since the quality variance becomes less dominant in the cost function as $\lambda$ increases. The same phenomenon is also observed in the CQ CBR encoding algorithms (see Section 5.4.3).

# Chapter 7

# Conclusions

## 7.1. Summary of Research Contributions and Publications

The following list summarizes the contributions of this thesis. For each contribution, related publications are also listed.

- *Development of the Laplacian Rate/Distortion Model for MPEG-4 Video Encoding*:

  The Laplacian rate/distortion (R/D) model for MPEG-4 video encoding is developed in Section 3.1 and published in [108] (for intra frame only) and [99] (for both intra and inter frames). This model is developed based on assuming the DCT (discrete cosine transform) coefficients of the original sequence have Laplacian distributions. Except the SQBCBR algorithm proposed in Section 4.2, this model is used by all encoding algorithms proposed in this thesis. Simulation results suggest that this proposed R/D model predict the number of bits and distortion quite accurately. Also, since temporal stationarity is not required nor assumed, this R/D model can deliver consistent performance for both stationary and non-stationary sequences.

- *Development of the Laplacian Rate/Distortion Model for MPEG-4 Video Transcoding*:

Since the transcoder input contains quantized DCT coefficients, which are requantized by the transcoder, the Laplacian R/D model for encoding cannot be applied to transcoding directly. However, by using the same assumption as in the encoding case, the Laplacian R/D model for MPEG-4 video transcoding is proposed in Section 3.2. It is used by all CQ (constant quality) transcoding algorithms proposed in Chapter 6. Due to the requantization effect, unlike in the encoding case, the transcoding R/D model has only numerical solution. However, as indicated in the thesis, the computational complexity is not great. Also, since the Laplacian parameters are estimated from quantized coefficients (and many of them are quantized to zero), the transcoding R/D model is less accurate than the encoding model. Despite this, the proposed R/D model can still give prediction accurate enough for the rate-control algorithm. This contribution has been submitted to [119].

- *Development of six Constant Quality (CQ) rate control algorithms for MPEG-4 video encoding:*

  In Chapter 5, a total of six CQ rate control algorithms are developed for MPEG-4 video encoding. They provide solutions for three CQ problems with different channel requirements: VBR (variable bit rate) with FTQ (fixed target quality), CBR (constant bit rate) with FTQ (published in [99] and [118]), and CBR with DTQ (dynamic target quality) (published in [115]). For each CQ problem, two algorithms are proposed that control the quality either at the frame level (using the Laplacian R/D model), or at the macroblock level (using real distortions). Unlike all previously reported CQ algorithms that control the quality indirectly through

target bit allocation, the proposed CQ algorithms controls the quality directly through the quality matching (QM) algorithms. Simulation results indicate that the FTQ algorithms can meet the target quality quite accurately while having a lower PSNR (peak signal-to-noise ratio) variance than Q2 and TM5 (Test Model 5). For the DTQ algorithms, an extra degree of control is offered ($\lambda$) that allows users to change the emphasis from having lower PSNR variance to meeting the target bitrate more accurately. Simulation results indicate that the DTQ algorithm can give a lower PSNR variance while having a similar average PSNR at a similar/lower bitrate when compared to Q2 and TM5.

- *Development of two MPEG-4 CQ rate control algorithms for MPEG-4 transcoding*:

  With the success of encoding CQ algorithms, the concept of direct quality control is further extended to MPEG-4 video transcoding. Two CQ DTQ transcoding algorithms are proposed in Chapter 6. They use the transcoding Laplacian R/D models to control the rate and quality. Simulation results indicate that the proposed CQ algorithms give lower PSNR variance and about 1~2 dB drop in average PSNR when compared with direct encoding using Q2 and TM5. This drop is expected due to the requantization error introduced by the transcoder. When the proposed algorithms are compared with the reference transcoding algorithms (Q2T and TM5T), simulation results indicate that proposed algorithms give lower PSNR variance and similar average PSNR. Simulation results indicate that the extra degree of control also works well in these algorithms. One

conference paper about this contribution has been submitted to [119]. Another journal paper [120] is under preparation that will be submitted in the near future.

- *Development of a MPEG-4 CBR rate-control algorithm using the Laplacian rate model*:

  A CBR rate-control (LPECBR) algorithm for MPEG-4 encoding is proposed in Section 4.1, and published in [108]. Although it uses the same target bit allocation algorithm as Q2 does, it uses the encoding Laplacian rate model to determine the QP (quantization parameter) to code the frame. As indicated by the simulation results, this algorithm offers a similar/higher PSNR and a lower variation in bitrate than Q2.

- *Integrating Watson's DVQ (digital video quality) metric into TM5 for MPEG-2 video encoding:*

  This contribution corresponds to the SQBCBR algorithm presented in Section 4.2. This algorithm was first presented in [109] and [121]. It is the only algorithm in this thesis for MPEG-2 encoding and uses Watson's DVQ (digital video quality) metric as the quality measure. Due to the low-pass filters used in Watson's DVQ, it gives better DVQ for low spatial activity sequences, but equal or worse DVQ for high spatial activity sequences. Since Watson's DVQ is not yet a standard way to predict the subjective quality, PSNR is still used as the quality measure in the rest of this thesis.

## 7.2. Conclusions

Based on the analysis and simulation results of this thesis, the following conclusions can be made:

- The Laplacian R/D models predict the number of bits/distortion quite accurately. The most important property is that the Laplacian model does not require the input to be temporally stationary, as implicitly assumed in many previously reported R/D models.
- The concept of controlling CQ directly is more intuitive and seems more efficient than previously reported CQ algorithms that control quality indirectly through target bit allocation. This concept is not only applicable for video encoding, but can also be extended for video transcoding.
- Although the MVQM algorithm gives suboptimum selection of macroblock QPs, simulation results indicate that it matches the target quality with a very high precision. It works by breaking a global target (target quality in this case) into smaller local targets for each stage of trellis. The Viterbi algorithm is used to find the minimum path in the trellis, where the distance between nodes is the absolute difference between the node value (macroblock distortion) and the local target. Note that the same approach can also be extended for other applications where a best matched distance path is needed.

## 7.3. Future Research Directions

Some recommendations and possible improvements for future research endeavors are discussed below.

- *Define a better measure of CQ*:

  In this thesis, PSNR variance is used as the measure of CQ. It is computed from the PSNR of the entire video sequence. However, since the encoding difficulties change over time, the target quality may also need to be adjusted over time. In such case, variance over the entire sequence may not be a good measure. As an example, assume that the PSNR curve is composed of two piecewise constant lines. From a user's point of view, there is only one sudden change in the entire video. Although the quality is constant in each segment of the video, if variance is measured from the entire sequence, it would probably be large, and would not reflect the true quality variation perceived by viewers. Thus, a better measure of CQ is necessary. As an example, it would be better to measure CQ as the average PSNR measured from a fixed-size sliding window. If this measure is used, the transition effect would not dominate the overall CQ measure. Another possible way to define CQ is to guarantee the minimum quality. From user's point of view, slightly poorer average (but nearly constant) quality is better than higher average quality but with occasional low qualities. These occasional low qualities could give bad impression to the viewers. Thus, it would be nice that the CQ measure could ensure a minimum quality as well as minimizing quality variations.

- *Use other objective quality measures*:

  The main quality measure used in this thesis is PSNR. Although it is the most commonly used quality measure, it is well known that PSNR does not necessarily reflect the perceptual quality in some situations. As discussed in Section 5.1.5, the proposed CQ algorithms are capable to use other quality measures as long as it has the additive property. This means that if other quality measure could give a better prediction of perceptual quality than PSNR, the performance of the proposed CQ algorithms can also be improved with little or no effort.

- *Use region-of-interest (ROI) in the MVQM and MVQMT algorithms*:

  In the MVQM and MVQMT algorithms, the actual macroblock MSEs (mean squared errors) are used. The distortions of all macroblock are treated equally. However, from end-user's point of view, not all regions in a frame are equally important. There is usually one or more ROI that is important to the viewers. Thus, it would be nice to emphasis the distortions in the ROI and deemphasize distortions for the rest. One possible way to achieve this is to scale the macroblock MSEs based on their importance.

- *Extend the concept of CQ control to joint coding and joint transcoding*:

  In the thesis, the concept of CQ control has been extended from encoding to transcoding. However, it can be further extended to joint coding and joint transcoding. Joint coding is used to transmit multiple sequences over a single communication channel, such as satellite communalization. Joint transcoding is joint coding with transcoding, i.e., multi-program transmission with format conversion. To extend the CQ DTQ algorithms to work from single sequence to

multiple sequences, it is essential to consider all sequences simultaneously. To be more specific, one would need to modify Step 1 (set the target quality) and Step 4 (prevent buffer overflow) of the algorithm presented in Section 5.4. One way to set the target qualities simultaneously is to minimize a new cost function. This new cost function needs to maximize the PSNRs and buffer usage while minimizing the PSNR variances. This could be done by adding a new sample variance term for each sequence into the cost function. As for Step 4, since the buffer is shared by all sequences, buffer overflow will also need to be prevented simultaneously. Thus, if buffer overflow is expected to happen after adding *all* current frames of multiple sequences and removing the average number of bits per frame (leaky-bucket), the remaining bit budget ($Bit_{Target}$, defined in equation (53)) will need to be split to all sequences. One way to split the bit budget is to use the relative magnitude of target quality. Thus, sequences with higher target quality will receive more bit budget than sequences with lower target quality. Bit matching algorithm can then be used to reset the QP(s) for each sequence.

# References


[1] ISO/IEC 13818-2:2000, *Information technology – Generic coding of moving pictures and associated audio information: Video*, 2000.

[2] ISO/IEC 14496-2:2001, *Information technology – Coding of audio-visual objects – Part 2: Visual*, 2001.

[3] A.B. Watson, J. Hu, J.F. McGowan III and J.B. Mulligan, "Design and performance of a digital video quality metric", *Human Vision and Electronic Imaging IV, Proceedings*, Volume 3644, B.E. Rogowitz and T.N. Pappas eds., pp. 168 – 174, SPIE, Bellingham, WA, 1999.

[4] A.B. Watson, "Toward a perceptual video quality metric", *Human Vision, Visual Processing, and Digital Display VIII*, 3299, pp. 139-147, 1998.

[5] M. Jiang, X. Q. Yi, and N. Ling, "Frame layer bit allocation scheme for constant quality video", *ICME 2004*, vol. 2 (2004) 1055-1058.

[6] J. Q. Lan, X. H. Zhuang, and W. J. Zeng, "Single-pass frame-level constant distortion bit allocation for smooth video quality", *ICME 2004*, vol. 1 (2004) 387-390.

[7] T. Chiang, and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model", *IEEE Trans. on CSVT*, vol. 7, no. 1, Feb. 1997, pp. 246-250.

[8] ISO/IEC-JTC1/SC29/WG11, MPEG93/457, *Test Model 5*, April 1993

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

[10] The MPEG Home Page, *http://mpeg.telecomitalialab.com*

[11] ISO/IEC 11172-2, *Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video*, 1993.

[12] ITU-T Recommendation H.261, *Video CODEC for audiovisual services at px64 kbit/s,* 1993.

[13] ITU-T Recommendation H.263, *Video coding for low bit rate communication, Version 2*, 1998.

[14] ISO/IEC 14496-10 and ITU-T Rec. H.264, *Advanced Video Coding*, 2003.

[15] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons, Chichester, England, 2003.

[16] Leonardo Chiariglione, "Short MPEG-1description", *http://mpeg.telecomitalialab.com/standards/mpeg-1/mpeg-1.htm*

[17] Leonardo Chiariglione, "Short MPEG-2 description", *http://mpeg.telecomitalialab.com/standards/mpeg-2/mpeg-2.htm*

[18] Rob Koenen, "MPEG-4 Overview (V.18 – Singapore version)", *http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm*

[19] R. Koenen, "MPEG-4 multimedia for our time", *IEEE Spectrum*, vol. 36, no.2, pp. 26-33, 1999.

[20] DivX Home Page, *http://www.divx.com*

[21] XviD Home Page, *http://www.xvid.org*

[22] RealNetwork Home Page, *http://www.real.com*

[23] QuickTime Home Page, *http://www.apple.com/quicktime*

[24] José M. Martínez, "Overview of MPEG-7 Standard (version 5)", *http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm*

[25] Jan Bormans, "Keith Hill, MPEG-21 Overview", *http://mpeg.telecomitalialab.com/standards/mpeg-21/mpeg-21.htm*

[26] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard", *IEEE Communications Magazine*, vol. 36, issue 6, pp. 112 –119, 1998

[27] Y. Wang, Q.F. Zhu, "Error control and concealment for video communication: a review", *Proceedings of the IEEE*, vol. 86, issue 5, pp. 974-997, 1998

[28] S. Cie, P. Cosman, "Comparison of error concealment strategies for MPEG video", *Wireless Communications and Networking Conference (WCNC)*, vol. 1, pp. 329-333, 1999.

[29] V. Papadakis, W.E. Lynch, T. Le-Ngoc, "Syntax based error concealment", *IEEE International Symposium on Circuits and Systems (ISCAS'98)*, vol. 4, pp. 265-268, 1998.

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

[31] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard", *IEEE Trans. on Circuit and Systems for Video Technology (CSVT)*, vol. 11, no.3, pp. 301-317, 2001.

[32] M. van der Schaar, H. Radha, "A hybrid temporal-SNR fine-granular scalability for Internet video", *IEEE Trans. on CSVT*, vol. 11, no.3, pp. 318-331, 2001.

[33] M. Domanski, A. Luczak, S. Mackowiak, "Spatio-temporal scalability for MPEG video coding", *IEEE Trans. on CSVT*, vol. 10, no.7, pp. 1088-1093, 2000.

[34] M. J. Quinn, *Parallel Computing – Theory and Practice*, McGraw-Hill, p. 6, 1994.

[35] Y. He; I. Ahmad, M.L. Liou, "A software-based MPEG-4 video encoder using parallel processing", *IEEE Trans. on CSVT*, vol. 8, no.7, pp. 909-920, 1998.

[36] S.M. Bliandarkar, S.R. Chandrasekaran, "Parallel parsing of MPEG video", *International Conference on Parallel Processing 2001*, pp. 444-451, 2001.

[37] N.S. Jayant, P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, 1984.

[38] Y. Huang, X.H. Zhuang, and C.S. Yang, "Two block-based motion compensation methods for video coding", *IEEE Trans. on CSVT*, vol. 6, no. 1, pp. 123-126, 1996.

[39] V.G. Moshnyaga, "A new computationally adaptive formulation of block-matching motion estimation", *IEEE Trans. on CSVT*, vol. 11, no. 1, pp. 118-124, 2001.

[40] F.H. Cheng; S.N. Sun, "New fast and efficient two-step search algorithm for block motion estimation", *IEEE Trans. on CSVT*, vol. 9, no. 7, pp. 977-983, 1999.

[41] ISO/IEC 14496-1:2001/Amd 1:2001, *Extended BIFS*, 2001.

[42] T. Schoepflin, V. Chalana, D.R. Haynor, Y. Kim, "Video object tracking with a sequential hierarchy of template deformations", *IEEE Trans. on CSVT*, vol. 11, no. 11, pp. 1171-1182, 2001.

[43] T. Meier, K.N. Ngan, "Automatic segmentation of moving objects for video object plane generation", *IEEE Trans. on CSVT*, vol. 8, no. 5, pp. 525-538, 1998.

[44] M. Kim, Jae G. Choi, D.H. Kim, et. al, "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information", *IEEE Trans. on CSVT*, vol. 9, no. 8, pp. 1216-1226, 1999.

[45] C. Toklu, A.M. Tekalp, A.T. Erdem, "Semi-automatic video object segmentation in the presence of occlusion", *IEEE Trans. on CSVT*, vol. 10, no. 4, pp. 624-629, 2000.

[46] J. Guo; J.W. Kim; C.J. Kuo, "An interactive object segmentation system for MPEG video", *ICIP 1999*, vol. 2, pp. 140-144, 1999.

[47] P. Salembier, F. Marques, "Region-based representations of image and video: segmentation tools for multimedia services", *IEEE Trans. on CSVT*, vol. 9, no. 8, pp. 1147-1169, 1999.

[48] A. Nosratinia, "New kernels for fast mesh-based motion estimation", *IEEE Trans. on CSVT*, vol. 11, no. 1, pp. 40-51, 2001.

[49] I. Celasun, A.M. Tekalp, "Optimal 2-D hierarchical content-based mesh design and update for object-based video", *IEEE Trans. on CSVT*, vol. 10, no. 7, pp. 1135-1153, 2000.

[50] N. Brady, "MPEG-4 standardized methods for the compression of arbitrarily shaped video objects", *IEEE Trans. on CSVT*, vol. 9, no. 8, pp. 1170-1189, 1999.

[51] T. Meier, K.N. Ngan, "Video segmentation for content-based coding", *IEEE Trans. on CSVT*, vol. 9, no. 8, pp. 1190-1203, 1999.

[52] A. N. Netravali, B.G. Haskell, *Digital Picture Representation, Compression, and Standards*, Plenum Press, 1995.

[53] C. Podilchuk, R. Safranek, *Signal Processing: Coding of Speech, Audio, Text, Image and Video (Selected Topics in Electronics and Systems – Vol. 9)*, Editor N. Jayant, World Scientific, 1997.

[54] K. Rao, P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, and Applications*, Academic Press, 1990.

[55] W. H. Chen, C. H. Smith, and S. C. Fralick, "A Fast Computational Algorithm for the Discrete Cosine Transform", *IEEE Trans. on Communications*, vol. Com-25, no. 9, pp. 1004-1009, September 1977.

[56] W. Li, "A new algorithm to compute the DCT and its inverse", *IEEE Trans. on Signal Processing*, vol. 39, no. 6, pp. 1305-1313, June 1991.

[57] C. Loeffler, A. Lightenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11-multiplications", *ICASSP-89*, vol. 2, pp. 988 –991, 1989.

[58] "IEEE Standard Specifications for the Implementation of 8x8 Inverse Discrete Cosine Transform", *IEEE Std. 1180-1990*, March, 1991.

[59] Z. Xiong, K. Ramchandran, M.T. Orchard, Y.Q. Zhang, "A Comparative Study of DCT- and Wavelet-Based Image Coding", *IEEE Trans. on CSVT*, vol. 9, no.5, pp. 692-695, 1999.

[60] I. Moccagatta, M.Z. Coban, H.H. Chen, "Wavelet-based image coding: comparison of MPEG-4 and JPEG-2000", *Conference Record of the 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1178-1182, 1999.

[61] G.W. Xing; J. Li; S.P. Li; Y.Q. Zhang, "Arbitrarily shaped video-object coding by wavelet", *IEEE Trans. on CSVT*, vol. 11, no. 10, pp. 1135-1139, 2001.

[62] H. Lohscheller, "A subjective adaptive image communication system", *IEEE Trans. on Communications*, vol. T-COMM 32, pp. 1316-1322, 1984.

[63] C.E. Shannon, "A Mathematical Theory of Communication", *Bell Systems Technical Journal*, vol. 27, pp. 379-423 and 623-656, July 1948.

[64] DivX.com Glossary Terms, *http://www.divx.com/support/glossary.php*

[65] VQEG home page: *http://www-ext.crc.ca/vqeg/main.html*

[66] VQEG Phase I Final Report, April 2000: *http://ftp.crc.ca/test/pub/crc/vqeg/Final_Report_April00.doc*

[67] S. Winkler, "A perceptual distortion metric for digital color video", *Human Vision and Electronic Imaging IV, Proceedings*, vol. 3644, B.E. Rogowitz and T.N. Pappas eds. , SPIE, Bellingham, WA, pp. 175 – 184, 1999, .

[68] S. Wolf and M. Pinson, "In-service performance metrics for MPEG-2 video systems", *Proc. Made to Measure 98 - Measurement Techniques of the Digital Age Technical Seminar, International Academy of Broadcasting (IAB), ITU and Technical University of Braunschweig, Montreux, Switzerland*, November 1998.

[69] P. K. Kaiser, and R. M. Boynton, *Human Color Vision*, 2nd Ed., Optical Society of America, Washington DC, pp. 250-253.

[70] H. Peterson, A. J. Ahumada, Jr. and A. Watson, "An Improved Detection Model for DCT Coefficient Quantization", *SPIE Proceedings*, 1913, pp. 191-120, 1983.

[71] P.A.A. Assuncao, M. Ghanbari, "Buffer analysis and control in CBR video transcoding", *IEEE Trans. on CSVT*, vol. 10, no.1, pp. 83-92, 2000.

[72] A. Vetro, C. Christopoulos, and H.F. Sun, "Video Transcoding Architectures and Techniques: An Overview", *IEEE Signal Processing Magazine*, March 2003, pp. 18-29.

[73] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams", *Signal Processing: Image Communication* vol. 8 (1996) pp. 481-500, 1996

[74] P.A.A. Assuncao, M. Ghanbari, "Post-processing of MPEG2 coded video for transmission at lower bit rates", *ICASSP 1996*, vol. 4, pp.1998–2001, May 1996

[75] P.A.A. Assuncao, M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams", *IEEE Trans. on CSVT*, vol. 8, no.8, pp. 953-967, 1998.

[76] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital Video Transcoding", *Proceedings of the IEEE*, vol. 93, no. 1, January 2005, pp. 84-97.

[77] J. Youn, J. Xin, and M.T. Sun, "Fast video transcoding architectures for networked multimedia," *Proc. IEEE Int. Symp. Circuits and Systems*, Geneva, Switzerland, vol. 4, May 2000, pp. 25-28.

[78] J. Youn, M.T. Sun, and J. Xin, "Video transcoder architectures for bit rate scaling of H.263 bit streams", *Proceedings of the seventh ACM international conference on Multimedia*, 1999, pp. 243-250.

[79] H.Z. Sorial, W.E. Lynch, A. Vincent, "Estimating Laplacian parameters of DCT coefficients for requantization in the transcoding of MPEG-2 video", *International Conference on Image Processing 2000 (ICIP 2000)*, vol. 1, pp. 956-959, 2000.

[80] A. Vetro, H.F. Sun, Y. Wang, "Object-based transcoding for adaptable video content delivery", *IEEE Trans. on CSVT*, vol. 11, no.3, pp. 387-401, 2001.

[81] H.Z. Sorial, W.E. Lynch, A. Vincent, "Joint transcoding of multiple MPEG video bitstreams", *ISCAS 1999*, vol. 4, pp. 251-254, 1999.

[82] J.-C. Tsai, "Rate control for low-delay video using a dynamic rate table", *IEEE Trans. on CSVT*, vol. 15, no. 1, Jan. 2005, pp.133 - 137.

[83] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics, *IEEE Trans. on CSVT*, vol. 38, pp. 82–93, Jan. 1990.

[84] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. on CSVT*, vol. 6, pp. 12–20, Feb. 1996.

[85] W. K. Pratt, *Digital Image Processing*. New York, Wiley, 1978, ch. 10.

[86] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. PROC-68, no. 3, pp. 7–12, Mar. 1960.

[87] R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. on Communications*, vol. COM-31, no. 6, pp. 835–839, Jun. 1983.

[88] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. on Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[89] F. Muller, "Distribution shape of two-dimensional DCT coefficients natural images," *Electron. Letters*, vol. 29, no. 22, pp. 1935–1936, Oct. 1993.

[90] T. Eude, R. Grisel, H. Cherifi, and R. Debrie, "On the distribution of the DCT coefficients", *ICASSP 1994*, vol. 5, Apr. 1994, pp. 365–368.

[91] S. R. Smooth and R. A. Lowe, "Study of DCT coefficients distributions", *Proc. SPIE*, Jan. 1996, pp. 403–311.

[92] M. Barni, F. Bartolini, A. Piva, and F. Rigacci, "Statistical modeling of full frame DCT coefficients", *Proc. Eur. Signal Processing Conf. EUSIPCO 98*, vol. III, Sep. 1998, pp. 1513–1516.

[93] F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistical analysis of the 2D-DCT coefficients of differential signal for images," *Signal Processing: Image Communication*, vol. 4, pp. 477–488, 1992.

[94] J. Ribas-Corbera, and S. Lei, "Rate control in DCT video coding for low-delay communications", *IEEE Trans. on CSVT*, vol. 9, no.1, Feb. 1999, pp. 172-185.

[95] Z. He, Y. K. Kim, and S. K. Mitra, "Low-delay rate control for DCT video coding via $\rho$-domain source modeling," *IEEE Trans. on CSVT*, vol. 11, pp. 928–940, Aug. 2001.

[96] Z. He, and S.K. Mitra, "A unified rate-distortion analysis framework for transform coding", *IEEE Trans. on CSVT*, vol. 11, no. 12, December 2001, pp. 1221-1236.

[97] N. Kamaci, Y. Altunbasak, and R.M. Mersereau, Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models, *IEEE Trans. on CSVT*, vol. 15, no. 8, Aug. 2005 pp. 994 – 1006.

[98] R. Reiniger and J. Gibbs, Distribution of the two dimensional DCT coefficients for images, *IEEE Trans. on Communications*, vol. COMM-31 (June 1983) 835-839.

[99] Cheng-Yu Pai, and W. E. Lynch, MPEG-4 Constant-Quality Constant-Bit-Rate Controls, *Image and Video Communications and Processing 2005*, *Proceedings of SPIE*, vol. 5685 (2005).

[100] H. V. Poor, *An introduction to signal detection and estimation*, Springer-Verlag, 1988.

[101] B. Xie, and W. Zhen, "Sequence-based rate control for constant quality video", *ICIP 2002*, vol. 1 (2002) 77-80.

[102] S.Y. Lee, and A. Ortega, "Optimal rate control for video transmission over VBR channels based on a hybrid MMAX/MMSE criterion", *ICME 2002*, vol. 2 (2002) 93-96.

[103] D. Quaglia, and J.C. de Martin, "Delivery of MPEG video streams with constant perceptual quality of service", *ICME 2002*, vol. 2 (2002) 85-88.

[104] L. Zhao, J.W. Kim, and C.-C. J. Kuo, Constant quality rate control for streaming MPEG-4 FGS video, *ISCAS 2002*, vol. 4 (2002) 544-547.

[105] D. Qiao, and Y.F. Zheng, Dynamic bit-rate estimation and control for constant-quality communication of video, *Proc. of the 3$^{rd}$ world congress on intelligent control and automation*, vol. 4 (2000) 2506-2511.

[106] J. Choi, Distortion policy of buffer-constrained rate control for real-time VBR coders, *IEEE Trans. Image Processing*, vol. 8, no. 4 (April 1999) 537-547.

[107] J.I. Ronda, F. Jaureguizar, and N. Garcia, Buffer-constrained coding of video sequence of quasi-constant quality, *ICIP 1996*, vol. 3 (1996) 367-370.

[108] Cheng-Yu Pai and William E. Lynch, "MPEG-4 Rate Control Algorithm using Laplace Parameter Estimation", *IEEE International Conference on Multimedia and Expo (ICME 2003)*, pp. 241-244, July 6-9, 2003.

[109] Cheng-Yu Pai, and William E. Lynch, "Subjective-Quality-Based MPEG-2 Video Compression", *IEEE International Conference on Information Technology: Coding and Computing (ITCC 2002)*, pp. 416-420, April 8-10, 2002.

[110] VQEG Phase II Final Report, August 25, 2003: *ftp://ftp.its.bldrdoc.gov/dist/ituvidq/frtv2_final_report/VQEGII_Final_Report.doc*

[111] W.-N. Lie, C.-H. Tseng, and T.C.-I. Lin, "Constant-quality rate allocation for spectral fine granular scalable (SFGS) video coding by using dynamic programming approach", *ICME 2004*, vol. 1 (2004) 655-658.

[112] L. Overmeire, F. Verdicchio, J. Barbarie, P. Schelkens, and L. Nachtergaele, "Constant quality rate-control for video encoding based on activity segmentation", *ICIP 2004*, vol. 2 (2004) 1133-1136.

[113] A.J. Viterbi, and J.K. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, 1979.

[114] G.D. Forney, "The Viterbi algorithm", *Proc. IEEE*, vol. 61, pp. 268-278, March 1973.

[115] Cheng-Yu Pai, and W. E. Lynch, "MPEG-4 Constant-Quality Constant-Bit-Rate Control Algorithms", Signal Processing: Image Communication, vol. 21, no. 1, January 2006, pp. 67-89.

[116] R. E. Walpole, and R. H. Myers, *Probability and Statistics for Engineering and Scientists*, 5th ed., Macmillan, New York, 1993, pp. 204-205.

[117] J.S. Kim, J.G. Kim, K.O. Kang, and J. Kim, "A distortion control scheme for allocating constant distortion in FD-CD video transcoder", *IEEE ICME 2004*, vol. 1, pp. 161-164, 2004.

[118] Cheng-Yu Pai, and W. E. Lynch, "MPEG-4 Constant-Quality Constant-Bit-Rate Controls", poster session in SYTACom (Centre for Advanced Systems and Technologies in Communications) technology forum 2005.

[119] Cheng-Yu Pai, and W. E. Lynch, "Constant-Quality CBR Rate-Control Algorithms for MPEG-4 Video Transcoding", submitted to *IEEE ICIP 2006* in January 14, 2006.

[120] Cheng-Yu Pai, and W. E. Lynch, "Constant-Quality CBR Rate-Control Algorithms for MPEG-4 Video Transcoding", to be submitted to a journal.

[121] Cheng-Yu Pai, W. E. Lynch, and A. Vincent, "Subjective-Based Video Compression", poster session in *CITR 2001.*