Transcoding of MPEG Compressed Video

Hani Sorial

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

0-612-68213-7

ABSTRACT

Transcoding of MPEG Compressed Video

Hani Sorial

Many video services use pre-encoded video for the distribution of video programs to end users. The transmission of compressed video over channels with different capacities may require a reduction in bit rate if the transmission media has a lower capacity than the capacity required by the video bitstream, or when the communication network is congested. The process of converting a compressed video format into another compressed format is known as *transcoding*. This thesis addresses the specific transcoding problem of bitrate reduction of a previously compressed MPEG video.

Fully decoding a compressed video then re-encoding it at a lower bit rate, as a second generation video, has two disadvantages. First, it is not an efficient solution in terms of implementation complexity, delay and cost. Second, errors are introduced in the repeated compression/decompression of MPEG video, known as multigeneration. In this research, five mechanisms contributing to the continued degradation in multigeneration of MPEG video are identified: Pixel Domain Quantization (PDQ), Pixel Domain Clipping (PDC), Compression Control Parameters Variation (CCPV), Motion Vector Re-estimation (MVR) and Error Propagation due to Motion Compensation (EPMC). The degradation caused by each mechanism is illustrated and quantified by experiments.

Next, the research addresses transcoding of MPEG compressed video. Two methods to reduce the requantization errors in transcoding are proposed. The first method assumes Laplacian distributions for the original DCT coefficients. A Laplacian parameter for each coefficient is estimated at the transcoder from the quantized input DCT coefficients. These parameters are used in transcoding to improve the quality of the transcoded video. The second method, selective requantization, is based on avoiding critical ratios of the two cascaded quantizations (encoding versus transcoding) that either lead to larger transcoding errors or require a higher bit budget. The experimental results show that both methods improve the quality of the transcoded video.

Moreover, the thesis addresses the problem of multi-program video transmission over heterogeneous networks and provides a joint transcoder for transcoding multiple MPEG video bitstreams simultaneously. It is shown that joint transcoding provides better picture quality than independent transcoding of each sequence at a constant bitrate. Furthermore, joint transcoding minimizes the variation in picture quality between the sequences, as well as within each sequence. Consequently, joint transcoding results in a better utilization of the channel capacity.

To my parents, my sister and my brothers

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. William E. Lynch for his excellent guidance and advice throughout this research work. I am extremely grateful for his patience, support, and friendship.

I would also like to thank Dr. M. Omair Ahmad and Dr. Eugene I. Plotkin for their valuable input over the course of my studies.

Special thanks goes to Andre Vincent from the Communications Research Centre in Ottawa for his input and expertise during the CITR research project.

In addition, I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canadian Broadcasting Corporation (CBC), and the STENTOR Resource Centre Inc. (SRCI) for their support through the industrial postgraduate scholarship program.

This research was also supported by a grant from the Canadian Institute for Telecommunications Research (CITR) under the NCE program of the Government of Canada.

Finally, I would like to thank the examining committee Dr. M. Omair Ahmad, Dr. Eugene Plotkin, Dr. Andre Zaccarin, Dr. Tien Bui, and Dr. William Lynch for their valuable comments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

α	Laplacian parameter
b	buffer fullness
Cb	picture Chrominance/blue color difference component
Cr	picture Chrominance/red color difference component
D	distance moved by quantization (Euclidean distance)
d_k	kth decision level of the quantizer
e	root mean-square quantization error
ε	quantization error
ε_c	cascading error
$f(\cdot)$	probability density function
$L^{(i)}$	*ith* generation loss
μ	mean value
P	probability
$PSNR^{(i)}$	peak signal-to-noise ratio at the *ith* generation
$p(\mathbf{y})$	joint probability distribution of the samples y
Q	quantization step size
$q(\cdot)$	quantizer operator
$qscale$	quantization scaling parameter
r_k	kth reconstruction level of the quantizer
R	bitrate in bits/sec
S	number of bits generated by coding a picture
σ	standard deviation
T	target number of bits

t	picture type
V	image data vector
Var	variance
Y	picture luminance component

LIST OF ACRONYMS

B-picture	Bidirectionally predicted picture
CBR	Constant Bit Rate
CDF	Cumulative Distribution Function
CCPV	Compression Control Parameters Variation
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DVD	Digital Video Disc
EPMC	Error Propagation due to Motion Compensation
GOP	Group Of Pictures
HDTV	High Definition Television
IDCT	Inverse Discrete Cosine Transform
ITU	International Telecommunication Union
ISO	International Organization for Standardization
I-picture	Intra picture
LCM	Least Common Multiple
MAD	Mean Absolute Difference
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
MVR	Motion Vector Re-estimation
NTSC	National Television System Committee
P-picture	Predicted picture
pdf	Probability Density Function
PAL	Phase Alternating Line

PDC	Pixel Domain Clipping
PDQ	Pixel Domain Quantization
PSNR	Peak Signal-to-Noise Ratio
RLC	Run-Length Coding
SDTV	Standard Definition Television
SECAM	Sequentiel Couleur avec Memoire
SIF	Source Input Format
TM5	Test Model document, version 5
VBR	Variable Bit Rate
VLC	Variable Length Coding
VLD	Variable Length Decoding

Chapter 1

Introduction

1.1 Introduction

Many video services use pre-encoded video for the distribution of video programs to end users. The transmission of compressed video over channels with different capacities may require a reduction in bitrate if the transmission media has a lower capacity than the capacity required by the video bitstream, or when the communication network is congested. The process of converting a compressed video format into another compressed format is known as *transcoding*. This thesis addresses the specific transcoding problem of bitrate reduction of a previously compressed MPEG video. Thus, throughout the thesis, the word transcoding will denote rate conversion of compressed video.

There are different approaches to bitrate reduction of compressed video including frame dropping, reducing the spatial resolution of the video, discarding high-frequency DCT coefficients, requantizing the DCT coefficients, and full decoding/re-encoding of the video. Usually, it is more advantageous to perform the rate conversion in the compressed domain, using partial decompression of the video bitstream. This reduces the computational complexity and delays as compared to fully decode the video and re-encode it at a lower bitrate. The former approach is adopted in this thesis.

Furthermore, in situations where a compressed MPEG video is subject to full decoding/re-encoding cycles, known as *multigeneration* of the video [1, 2], and in addition to the computational complexity involved in the MPEG encoding algorithm, the picture quality degrades at each cycle even though there is no manipulation of the image data and the compression factor is kept the same in each cycle. Errors in multigeneration of MPEG video result from different mechanisms associated with the MPEG encoding algorithm. This thesis identifies those mechanisms and quantifies the degradation caused by each mechanism [3, 4].

In this thesis, bit-rate conversion of MPEG compressed video is achieved by partial decompression of the video bitstream, then requantizing the DCT coefficients. Rate conversion of the compressed video involves two subsequent quantizations: the first in encoding, and the second in transcoding. In general, cascaded quantizations lead to extra degradation in the video quality as compared to direct quantization of the original video with the second quantizer. However, it is possible to reduce the extra errors by providing requantization methods for efficient transcoding [5, 6]. This thesis presents two methods to reduce the requantization errors in transcoding [7, 8].

Moreover, in a multi-program video transmission environment, when several video sequences are transmitted over a fixed communication channel, transcoding may be necessary if the channel has a lower capacity than the bandwidth required by these sequences. A straightforward approach would be to transcode each sequence at a constant bitrate such that their total bitrate meet the channel capacity. However, since scene complexities among different programs usually vary significantly with time, joint transcoding of these sequences using a joint bit-allocation between sequences according to their relative scenes complexity is more efficient in terms of overall output video quality than independent transcoding of each sequence at a constant bitrate. In general, while independent coding of multiple programs at a fixed rate may lead to large variations in picture quality between the programs, recent studies has shown that joint encoding achieve a more uniform picture quality and is more effective for compression of multiple video programs [9, 10, 11, 12]. This thesis extends this approach and provides a joint transcoder for transcoding multiple MPEG compressed sequences transmitted over a fixed communication channel [13].

1.2 Organization of the Thesis

This thesis is organized as follow:

Chapter 2 covers background material related to the work presented in this thesis. The chapter presents an overview of digital video compression standards and of the common methods used by these standards to achieve compression such as transform coding, motion compensated prediction, and entropy coding. Furthermore, The chapter discusses other methods used to flexibly support different transmission bandwidth and receivers with different display capabilities, such as layered coding. Next, A brief discussion of object-based coding, a technique that provides an object-layered representation allowing coding, access and manipulation of individual audio-visual objects in the scene is provided. The chapter further presents various figures of merit used to evaluate the performance of video compression systems including subjective and objective measures of picture quality, and factors that influence the video quality such as visual complexity, encoding parameters, and multigeneration of video. Other factors that affect the performance of video compression systems such as encoding and decoding delays, robustness and implementation complexity are briefly discussed. Finally, various picture impairments that may occur in a video compression system are presented.

In situations where bitrate conversion of MPEG compressed video is necessary, a straightforward approach is to decompress the pre-encoded video then to re-compress it as a second generation video at a lower bitrate using a cascaded decoder/encoder system. In addition to complexity, this approach introduces multigeneration errors in picture quality due to the repeated compression/decompression of MPEG video. Chapter 3 studies the errors that result from multigeneration of

MPEG video and identifies the mechanisms that cause degradation in video quality. Experiments are provided to illustrate and quantifies the errors introduced by each degradation mechanism. This chapter is a part of the contributions of this thesis [3, 4].

Chapter 4 addresses the specific transcoding problem of bit-rate reduction of a previously compressed MPEG video. The chapter starts by presenting the different approaches to bit-rate conversion of MPEG video, contrasting transcoding to other rate conversion approaches such as scalable coding and full decoding/re-encoding of the video. The chapter discusses the importance of transcoding in providing flexibility in transmission bit-rates in situations such as network congestion, or when the transmission channel has a lower capacity than the capacity required by the pre-encoded video. Next, the chapter presents issues related to transcoding that will be addressed throughout the thesis such as transcoder structure, requantization and rate control.

An important issue in bit-rate conversion of compressed video is to provide requantization methods for efficient transcoding. Chapter 5 proposes two methods to reduce requantization errors in transcoding. The first method, assumes Laplacian distributions for the original DCT coefficients. A Laplacian parameter for each coefficient is estimated at the transcoder from the quantized input DCT coefficients. These parameters are used to reconstruct the original DCT coefficients prior to the requantization process at the transcoder. The second method, selective requantization, is based on avoiding critical ratios of the two cascaded quantizations (encoding versus transcoding) that either lead to larger transcoding errors or require a higher bit budget. Experiments are provided at the end of this chapter to illustrate the improvement achieved in picture quality by using each requantization method. This chapter is a part of the contributions of this thesis [7, 8].

Chapter 6 discusses the problem of multi-program video transmission over a fixed communication channel and addresses the situations where the communication channel has a lower capacity than the capacity required by the video bitstreams. The chapter presents a joint transcoder for transcoding multiple MPEG video bitstreams simultaneously and shows that joint transcoding reduces the variation in picture quality between the video sequences and within the individual sequences, as compared to independently transcoding each sequence at a constant bitrate. An important issue in joint transcoding is the joint bit-allocation method. The chapter provides detailed formulation on the joint rate control used in this work. The chapter also presents a joint bit-allocation method to implicitly minimize the quality variation between the jointly transcoded sequences. Moreover, the interaction of the joint transcoder with channel traffic is discussed. A comparison between joint transcoding and independent transcoding concludes that joint transcoding outperforms independent transcoding, even in the presence of channel traffic. Experimental results support the conclusions provided in this chapter. This chapter is a part of the contributions of this thesis [13, 12].

Finally, Chapter 7 concludes the thesis and summarizes the contributions provided in this work. The chapter also discusses future research directions in transcoding of compressed video.

Chapter 2

Digital Video Compression Standards: An Overview

2.1 Digital Video Compression Standards

Video sequences contain a large amount of data and require large storage capacity and transmission bandwidth. However, this data contains considerable *redundancies* and therefore compression is possible. The main goal of video compression is to offer savings in transmission and storage resources.

Digital video is compressed by reducing the redundancies in both spatial and temporal directions [14, 15, 16]. Spatial redundancy is expressed by the existing correlation between neighboring pixels in one frame, while temporal redundancy is represented by the correlation between consecutive frames in the sequence.

Furthermore, the Human Visual System (HVS) is less sensitive to coding errors in higher frequencies than those in lower frequencies [14, 17]. Thus compression techniques can take advantage of this psycho-visual property so that picture information corresponding to higher frequencies is quantized more coarsely, and therefore a lower bitrate can be achieved.

Many standards for digital video compression exist and cover a wide range of applications including multimedia CD-ROMs, TV broadcasting, interactive distance education and entertainment, video conferencing, and multimedia database services.

The International Telecommunication Union (ITU) developed H.261 and H.263 standards [18, 19] for audiovisual services such as video conferencing. H.261 was designed for data rates which are multiples of 64Kb/s. These data rates (64-1920 Kb/s) suit ISDN (Integrated Services Digital Networks) lines. The H.263 was developed for low bitrate communication, with emphasis on bitrates below 64Kb/s.

The International Organization for Standardization (ISO) established the Moving Picture Experts Group (MPEG) in 1988 to generate standards for digital video and associated audio. MPEG developed the MPEG-1 and MPEG-2 video coding standards in 1992 and 1994, respectively [20, 21]. MPEG-1 was intended for coding of moving pictures and associated audio for multimedia and CD-ROM applications at up to about 1.5 Mbit/s. MPEG-2 was designed to be backward compatible to MPEG-1. It provides higher quality video (4-300 Mbit/sec) necessary for applications such as HDTV (High Definition Television) and SDTV (Standard Definition Television) broadcasting, TV production, and DVD (Digital Video Disc). MPEG-2 is also known as ITU-T Recommendation H.262 [21].

MPEG has finalized the MPEG-4 standard [22, 23] in October 1998. Unlike the above standards which deals with "frame-based video", MPEG-4 however codes objects separately. It is designed to provide an efficient coding of video over a wide range of bitrates, to offer a new kind of interactivity, and to integrate objects of different natures (e.g. natural video, graphics and text). Its targeted applications are mainly Internet multimedia, interactive video, video conferencing, videophone, wireless multimedia, and database services over ATM (Asynchronous Transfer Mode) networks [24].

MPEG is developing another standard, MPEG-7, targeted for July 2001. MPEG-7 is concerned with the standardization of a "Multimedia Content Description Interface" which will allow the description, identification and access of audiovisual information [25]. It is intended to provide complementary functionality to the other MPEG standards.

This chapter presents an overview on current digital video compression technology, with emphasis on the common methods used in video compression standards. The organization of this chapter is as follows. Section 2.2 presents common methods used in video compression standards, specifically, transform coding, motion compensated prediction, and entropy coding. Section 2.3 discusses scalable coding, a technique that allows for a layered representation of the coded bitstream. Scalable coding can be used as an alternative approach to bitrate reduction of MPEG compressed video in a prioritized Transmission such as ATM networks. In the case of network congestion, lower priority ATM cells may be discarded to reduce the bitrate of the video. Using this approach however for transmission of MPEG-2 compressed video can provide only a limited number of transmission bitrates as compared to transcoding which can flexibly provide a wide range of transmission bitrates. Next, Section 2.4 discusses various figures of merit used to evaluate the performance of video compression systems. Section 2.5 presents various picture impairments that may occur in a video compression system. Finally, Section 2.6 concludes this chapter.

2.2 Methods Used in Video Compression Standards

All video compression standards discussed above share common methods to achieve compression: a DCT (Discrete Cosine Transform) based method used to reduce the spatial redundancy within a video frame, and a Motion Compensated Differential Pulse Code Modulation (MC-DPCM) method used to exploit the temporal redundancy between frames. In addition to these lossy compression methods, run-length coding and variable length coding (VLC) are used to achieve further lossless compression.

Figure 2.1 shows a Motion Compensated DCT (MC-DCT) based encoder structure used in video compression standards [18, 19, 20, 21]. The encoding process may begin with some preprocessing which may include color conversion, format translation (e.g., interlaced to progressive), pre-filtering and subsampling. These

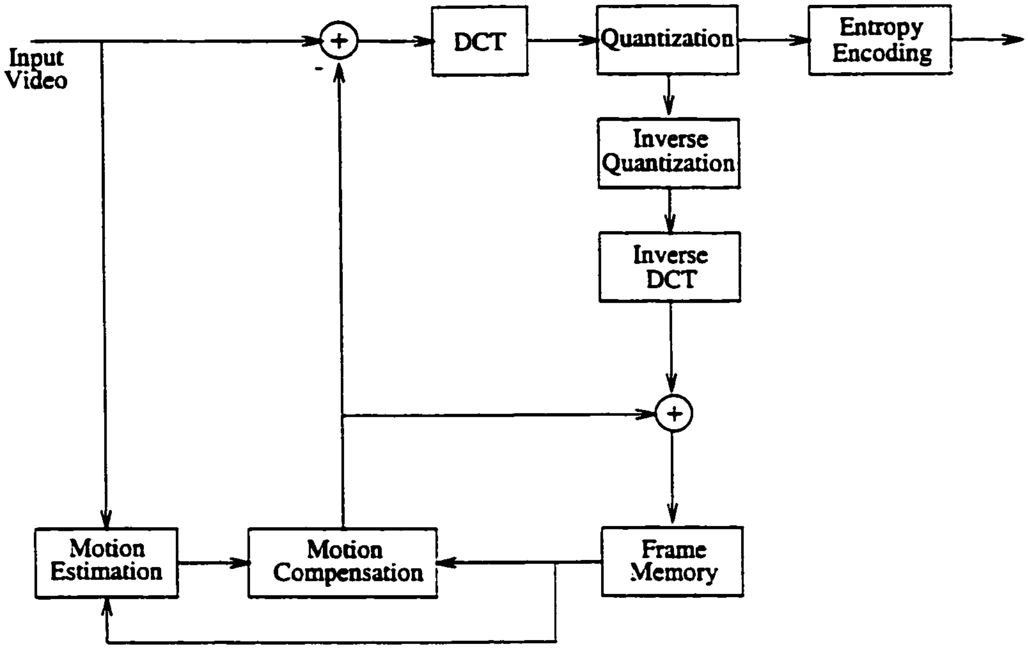


Figure 2.1: MC-DCT based encoder structure.

operations however are not specified in the standards [16].

Two types of coding are used: *intraframe* and *interframe*. In intraframe coding, the compression algorithms employ DCT coding techniques on 8 × 8 blocks to explore the spatial correlations between neighboring pixels. In interframe coding, a hybrid motion compensated prediction followed by transform coding of the remaining spatial information is used to achieve high data compression.

Next section describes block-based transform coding with emphasis on the discrete cosine transform which is used in current video compression standards.

2.2.1 Transform Coding

In transform coding [17, 26], a block of dependent pels (picture elements) is transformed into a set of less correlated coefficients. Usually, the transform is linear and orthogonal. The transform coefficients are then uniformly quantized to achieve compression. Further, the transformation usually provides energy compaction into few transform coefficients, which is a desirable property for signal compression [15].

The motivation for why transform coding has relatively good capability for bitrate reduction comes from the following:

- Not all transform coefficients need to be transmitted to maintain good image quality.

- Coefficients that are coded need not be represented with full accuracy.

In addition, as the transform coefficients are generally related to the spatial frequencies in the image, compression techniques can take advantage of the psychovisual property of the human visual system by quantizing more coarsely the higher-frequency coefficients.

In order to illustrate transform coding, consider the linear transformation of an $N \times N$ block of data x. Assuming a separable $2-D$ transformation, this can be written as:

$$X = A^T x A \tag{2.1}$$

where A is an $N \times N$ transform matrix also referred to as transformation kernels or basis functions, and X is an $N \times N$ matrix of transform coefficients. Note that A is an orthogonal matrix.

Lossy transform-based compression methods usually achieve compression by quantizing the transform coefficients and discarding those that are less important. If $\tilde{X}$ represents the $N \times N$ matrix of quantized coefficients at the receiver, then the reconstruction of x, denoted by $\tilde{x}$, is given by the inverse transform:

$$\tilde{x} = A\tilde{X}A^T \tag{2.2}$$

The optimum transform in a mean-square sense, is one that minimizes the mean square reconstruction error for a given number of total bits. This is the Karhunen-Loève Transform (KLT) [26, 27]. However, the basis functions of the KL transform are image dependent and of relatively high computational complexity. A solution to data dependency is to use statistical image models (e.g. first order Autoregressive model AR(1) [26]).

The discrete cosine transform is seen to be close in performance to the KLT, with the advantage that its basis functions are image independent. In practice, the DCT is widely used and is the basis of all image and video compression standards.

DCT-based Coding

The choice of the DCT [17, 28, 29, 30] in video compression standards comes from the important benefits it offers:

- The compaction efficiency of the DCT is close to that of the KLT for images with high inter-pixels correlation coefficient. It was shown that for a first order Autoregressive signal (AR(1)), as the correlation coefficient between adjacent samples approaches one, the basis functions of the KLT become the basis functions of the DCT [31].
- The DCT basis is image independent. This avoids the additional complexity needed to determine the basis, as well as the requirement of constant updating for non-stationary signals [15].
- Computations of the DCT can be performed using fast algorithms. This is desirable from both hardware and software implementation viewpoint.
- The DCT is an orthogonal transform. The two-dimensional DCT of an $N \times N$

data block is defined as:

$$X(u,v) = \frac{2}{N}C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(i,j)\cos\frac{(2i+1)u\pi}{2N}\cos\frac{(2j+1)v\pi}{2N} \qquad (2.3)$$

$$i,j,u,v = 0,1,2,\ldots,N-1$$

and

$$C(u),C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u,v=0 \\ 1 & \text{otherwise} \end{cases}$$

where $x(i,j)$ are the data values and $X(u,v)$ are the transform coefficients.

The inverse DCT is given by

$$x(i,j) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)X(u,v)\cos\frac{(2i+1)u\pi}{2N}\cos\frac{(2j+1)v\pi}{2N} \qquad (2.4)$$

Note that the transformation kernels are separable. Typically, a two-dimensional DCT is performed on 8×8 image blocks in two steps, by applying a one-dimensional DCT first to the rows then to the columns. More details on the DCT can be found in [28, 29, 30].

Next section reviews the basics of predictive coding.

2.2.2 Predictive Coding

In order to describe predictive coding, consider first Pulse Code Modulation (PCM), the most basic compression method. PCM consists of sampling an analog signal and quantizing each sample using a scalar quantizer of finite set of quantization levels. PCM quantizes the data independently. However, in order to get greater compression, the dependencies between neighboring pixels in space and time need to be exploited. A well known method that takes advantage of data dependencies is differential PCM (DPCM).

DPCM

The basic idea of DPCM [26] is to estimate the present data value based on previously encoded data. The prediction error (the difference between the actual sample and the predicted sample) is then coded and transmitted.

To illustrate DPCM, consider the simplest case where the prediction is based on the previous M quantized samples of an input image. Assuming a linear prediction, this can be expressed as:

$$\hat{x}(n) = \sum_{m=1}^{M} a_m \tilde{x}(n-m) \tag{2.5}$$

where $\hat{x}(n)$ is the predicted value of $x(n)$, a_m correspond to the prediction filter coefficients, and $\tilde{x}(n-m)$ are the previous quantized samples. The prediction error $d(n)$ is defined as:

$$d(n) = x(n) - \hat{x}(n) \tag{2.6}$$

and the prediction gain [26] is given by:

$$G_p = \frac{{\sigma_x}^2}{{\sigma_d}^2} \tag{2.7}$$

where σ_x and σ_d are the variances of the input signal and the prediction error signal, respectively.

Next section presents motion compensated prediction, a main coding method in video compression standards [14].

2.2.3 Motion Compensated Prediction

Motion compensation uses the same idea as the above, but the prediction is more complicated. Here, a temporal prediction is formed using temporally adjacent samples; which are determined through the process of motion estimation [16, 32].

Motion-compensated prediction takes advantage of the strong correlations between consecutive frames in a sequence. A current frame could be predicted from a past reference frame (*forward* prediction) and/or a future reference frame (*backward* prediction).

There are three main picture types that may be used in video compression. *Intra* coded pictures (I-pictures) are coded without reference to other pictures (i.e., without motion compensation). They are useful to provide access points in the coded sequence. I-pictures are coded with a relatively moderate compression. *Predictive* coded pictures (P-pictures) are coded more efficiently than I-pictures using motion compensated prediction. They may be predicted from past I or P frames and are used as references for further prediction. P-pictures can propagate coding errors. *Bidirectionally* predicted pictures (B-pictures) provide the highest degree of compression. They are coded using motion compensated prediction from either the past and/or future I or P pictures. B-pictures do not propagate errors because they are never used as reference pictures. They are also useful in predicting uncovered areas that do not appear in previous pictures.

Note that in video compression standards, the organization of picture types is flexible and depends on the application.

Motion Estimation

One of the most computationally intensive operations in video compression is the motion estimation process involved in a motion compensated prediction coding system [16]. In a real scene, the motion is complex. However, most motion compensation techniques use simple block-based motion estimation methods [15, 16]. These methods assume that the motion is *translational* and moves in a plane that is parallel to the camera plane, therefore excluding zoom and rotational motion. They also

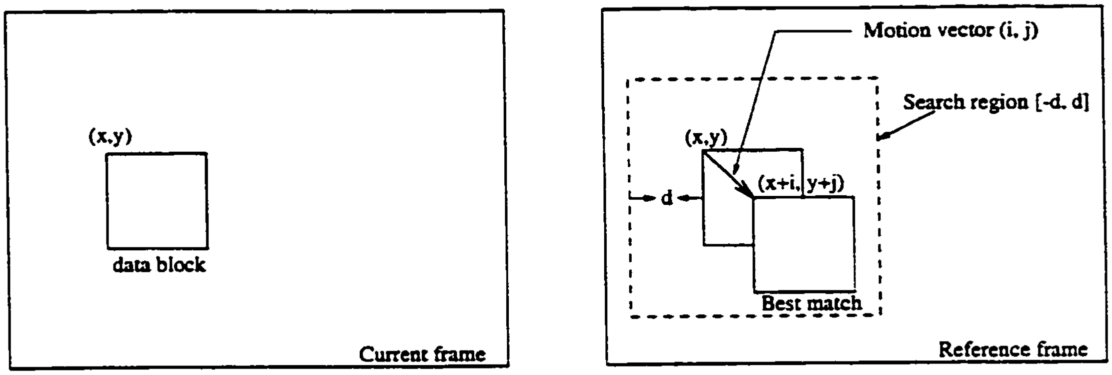


Figure 2.2: Block-based motion estimation process.

assume that the illumination is spatially and temporally constant so that matching can be done based on pixel intensity between frames. Further, it is assumed that occlusion of one object by another [33], as well as self-similarity of objects are relatively small.

Figure 2.2 illustrates the block-based motion estimation process. A current frame is divided into blocks. The location of each block is given by the (x, y) coordinates of its left-top corner. To predict a block in a current frame, a search region $[-d, d]$ around the original location of the block is identified in the reference frame. This region is then searched for the best matching block. Let $(x + i, y + j)$ be the location of the best match. The vector from (x, y) to $(x + i, y + j)$ is referred to as the *motion vector* associated with that block. Typically, the motion vector is expressed in relative coordinates, that is; it is simply expressed as (i, j).

Note that block-based motion compensation methods assume that the motion is constant within a block. The smaller the block size is, the more valid this model becomes, but with increasing overheads for the transmission of motion vectors [15, 16]. Block-based motion compensation is implemented in all current video compression standards [15].

Best-match Criterion

For each block in a current frame, the corresponding best matching block in the reference frame is calculated by minimizing a cost function. Two widely used cost functions [34] are the Mean Square Error (MSE) defined as:

$$MSE(i,j) = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}[S_c(x+m,y+n) - S_r(x+m+i,y+n+j)]^2 \quad (2.8)$$

and the Mean Absolute Difference (MAD) defined as:

$$MAD(i,j) = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}|S_c(x+m,y+n) - S_r(x+m+i,y+n+j)| \quad (2.9)$$

where S_c and S_r are $M \times N$ blocks in current and reference frames, respectively; and $-d \leq i,j \leq d$.

The best match is given by the coordinates (i,j) for which the cost function is minimized. These coordinates also define the motion vector. The MAD is more attractive than the MSE due to the reduced computational complexity [15].

Search Algorithms used in Block-based Motion Estimation

In order to find the best match with a maximum displacement of d, an exhaustive search would require $(2d+1)^2$ calculations of the cost function. Generally, a full search is computationally expensive but guarantees finding the optimum match. Several techniques have been proposed in order to reduce the computational complexity involved in the search. Three known techniques are the 2D-logarithmic search, the three-step search and the conjugate direction search. More details on these techniques can be found in [34]. Note that these techniques are heuristic and do not guarantee to find an optimum match.

Since the true displacements between frames may not be at integer pixel resolution, fractional pixel accuracy could result in a better motion compensated prediction. This is achieved by interpolating the frames before applying motion compensation. MPEG video compression standards permit motion vectors to be specified at least to a half-pixel accuracy [16].

2.2.4 Quantization

In video compression standards, the DCT transform is performed on 8×8 blocks of data. Equation 2.3 gives the two-dimensional DCT coefficients $X(u, v)$. As stated earlier, these coefficients are related to the spatial frequency content of the data block. The coefficient $X(0,0)$ is known as the *DC* coefficient of the block and is the average of the input block samples. Other coefficients are known as the *AC* coefficients of the transform.

The 64 DCT coefficients are uniformly quantized. The quantizer step size that is used for each coefficient is specified in a quantization table. This table is part of the information that must be transmitted to the decoder. The quantization process [15] is defined as a division of each DCT coefficient $X(u, v)$ by its corresponding entry from the quantization table $Q(u, v)$, followed by rounding to the nearest integer.

$$X_Q(u, v) = Round\left(\frac{X(u, v)}{Q(u, v)}\right) \tag{2.10}$$

At the decoder, inverse quantization [15] is performed as

$$\tilde{X}(u, v) = X_Q(u, v)Q(u, v) \tag{2.11}$$

A set of quantization tables that are derived from psycho-visual experiments [35] are given in MPEG-2 standards [21].

Next section presents the different coding decisions for macroblocks.

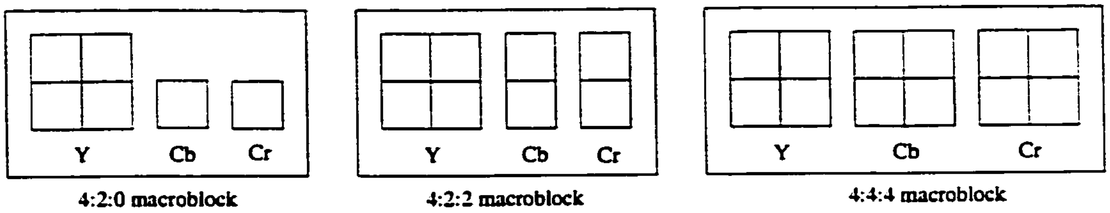


Figure 2.3: Different macroblock structures.

2.2.5 Macroblock Coding Modes

In block-based MC-DCT coding standards, motion compensation and coding decisions are performed on a *macroblock* basis. A macroblock is a unit (typically 16×16) that contains blocks of data from the luminance component (Y component) and from the spatially co-sited chrominance components (Cb and Cr components) [21]. There are three chrominance formats for a macroblock, namely, 4:2:0, 4:2:2, and 4:4:4 formats. These formats are shown in Figure 2.3.

Within a single I, P, or B picture, macroblocks can be coded differently. I-pictures are coded without motion compensation. Each macroblock may be coded using a different quantizer scale ($Mquant$). In general, the encoder may decide to change $Mquant$ to prevent possible buffer overflows or underflows.

The encoding decision for macroblocks in P and B pictures are summarized in the following:

- Decide if you want to code the macroblock as intra-type or inter-type macroblock. In some cases it is better to code a macroblock in P and B pictures as an I-macroblock. This may happen if motion estimation fails due to a high temporal activity.

- Decide if a macroblock need to be coded or not. If after quantization, all coefficients in the macroblock are zero, then the macroblock is skipped.

- Decide if some of the blocks within a macroblock may be skipped. A coded block pattern is used to indicate which of the blocks within a macroblock are coded.
- Decide if the quantizer scale needs to be changed.

More details on macroblock coding can be found in [16].

2.2.6 Entropy Coding

Entropy coding is the last stage in the encoding algorithms of video compression standards [18, 19, 20, 21]. It is a lossless compression stage following the quantization of the DCT coefficients. Entropy coding consists of two main steps: run-length coding (RLC) and variable-length coding (VLC).

Before entropy encoding, extra processing is applied to the DC coefficient $X(0,0)$. This coefficient represents the mean value of the input block. Usually there is a high correlation between DC coefficients of adjacent transform blocks. Thus, in order to take advantage of this correlation, the difference between the quantized DC coefficients of adjacent blocks is computed. This value is then coded instead of the original quantized value.

Run-length Coding

After quantization of the DCT coefficients, and since in general images tend to have low-pass spectrum, the non-zero DCT coefficients will tend to cluster at low frequencies and a large number of high frequency coefficients are likely to be zero. The quantized DCT coefficients are ordered in a *zig-zag* scan [16, 20, 21] such that non-zero coefficients will tend to be sent first. There will normally be a large run of zero coefficients at the end of the scan. An end-of-block marker is usually used to eliminate the need to transmit these coefficients.

Each AC coefficient is represented by its value and the run-length of zero valued coefficients that occur before it. The run/value combinations are mapped into codewords. Usually these codewords have a peaked distribution and are further compressed using VLC.

Variable-length Coding

Variable-length coding [17] is a lossless compression technique that can achieve a reduction in the average number of bits per codeword by assigning shorter codes to codewords having high probability of occurrence and longer codes to codewords having lower probability [17].

The average codeword length will be

$$C_{av} = \sum_{k} C_k P_k \tag{2.12}$$

where C_k denotes the kth codeword length and P_k is its probability of occurrence.

A fundamental result due to Shannon [36] establishes the entropy H of the source as a lower bound for the average number of bits per source symbol needed to code a discrete source, i.e.,

$$C_{av} \geq H \tag{2.13}$$

and

$$H = -\sum_{k} C_k log_2 P_k \tag{2.14}$$

as $\{P_k\}$ becomes highly concentrated, the entropy becomes smaller, and a variable length coding technique is more advantageous.

In video compression standards, the codewords representing the run/value combinations of the quantized DCT coefficients are usually coded using Huffman code [37, 38]. The code satisfies the *prefix* rule, which states that no code forms the prefix of any other, that is, the code is uniquely decodable once its starting point is known.

2.2.7 Rate Control

An important feature supported by video compression standards is the possibility of generating both constant and variable bitrates to suit different applications such as storage of video and real-time transmission. The rate control algorithm however is not part of the standards and is thus left to the implementers to develop efficient strategies.

In constant bitrate (CBR), the picture quality may vary depending on its content. Variable bitrate (VBR) however is meant to provide constant quality coding [21].

Rate control is based mainly on a buffering technique. An output buffer control the quantization of macroblocks depending on the buffer fullness. Many researchers proposed algorithms for bitrate control. One well known rate control algorithm is described in the Test Model document, version 5 ($TM5$) [39]. The algorithm consists of three main steps, a *target bit allocation* which estimates the number of bits available to code the next picture, a *rate control* that uses a "virtual buffer" to set the reference value of the quantization parameter for each macroblock, and an *adaptive quantization* which modulates the reference value of the quantization parameter according to the spatial activity in the macroblock in order to derive the actual value of the quantization parameter used to quantize the macroblock. More details on the $TM5$ rate control algorithm are available in [39].

More details on bitrate control can be found in [40, 41, 42].

2.3 Scalable Coding

Scalable coding [14, 16, 21, 43, 44, 33, 45] allows for a layered representation of the coded bitstream. It provides interoperability between different services and flexibly supports receivers with different display capabilities.

In scalable coding, it is also possible to assist concealment techniques by arranging the coded video information such that the most important information have a higher priority level in transmission [21]. A loss of less important information can then be effectively concealed. The main challenge in prioritizing the data layers is to reliably deliver video signals in the presence of channel errors, such as cell loss in ATM-based transmission networks [14]. In case of network congestion, scalable coding can provide a limited number of transmission bitrates. However, this approach is not as flexible as transcoding which can provide a wide range of bitrate reductions for MPEG compressed video.

Scalability allows a simple video decoder to decode and produce basic quality video while an enhanced decoder may decode and produce an enhanced quality video. This is possible because in scalable coding, the video is coded as two or more layers, an independently coded base layer and one or more enhancement layers coded dependently. Scalable coding offers a means of scaling the decoder complexity if processor and memory resources are limited. Scalability is implemented in MPEG-2 and MPEG-4.

There are four basic modes of scalable coding: data partitioning, SNR scalability, spatial scalability, and temporal scalability. These basic schemes can be combined to form a hybrid scalability scheme [16]. In the following, a two-layers scalable coding is assumed: a base layer and an enhancement layer.

- **SNR scalability:** provides two layers with the same spatial resolution but different video quality. the highest quality video is reconstructed by decoding both the base and enhancement layers. SNR scalability is based on frequency (DCT-domain) scalability technique. At the base layer, the DCT coefficients are coarsely quantized to achieve moderate image quality at reduced bitrate. The enhancement layer encodes the difference between the non-quantized DCT-coefficients and the quantized coefficients from the base layer with finer quantization stepsize.

  SNR scalability has been primarily developed for video applications that support multiple quality levels in prioritized transmission media.

- **Spatial scalability:** has been developed to support displays with different spatial resolutions. It provides two layers of different spatial resolutions. A lower spatial resolution video can be reconstructed from the base layer. The enhancement layer uses the spatially interpolated base layer to provide a video stream at the full spatial resolution. Spatial scalability is based on a classical pyramidal approach for progressive image coding [46, 47].

  Spatial scalability is useful for many applications including embedded coding for HDTV/SDTV systems, allowing a migration from SDTV services to higher spatial resolution HDTV services [14].

- **Temporal scalability:** allows layering of video frames. The base layer is coded by itself and provides the basic temporal rate. The enhancement layer is coded using temporal prediction with respect to the base layer. The two layers combined together give the full temporal resolution of the video.

Temporal scalability can be used in stereoscopic video for receivers with stereoscopic display capabilities [48, 49]. Layering can be achieved by providing a prediction of one of the images of the stereoscopic video (e.g., left view) in the enhancement layer based on coded images from the opposite view in the base layer [14].

- **Data partitioning:** splits the bitstream into two layers, called partitions. For example; the first partition may include address and control information and low-frequency DCT coefficients, while the second partitions contains the high-frequency DCT coefficients. The first partition may then be provided the highest transmission error protection.

  Data partitioning can be implemented with a very low complexity compared to other scalable coding schemes [14].

More details on scalable video coding can be found in [16, 21, 43, 44].

2.4 Figures of Merit

Various figures of merit may be used to evaluate the performance of video compression systems. Subjective and objective measures are used to rate the video quality of different video codecs (coder/decoder). Furthermore, by understanding the factors that influence the quality of video in a compression system, higher video quality can be achieved. Other important issues from a practical viewpoint are delays, robustness and error concealment, implementation complexity and cost.

Detailed discussion of the various figures of merit comes next.

2.4.1 Video Quality Measures

Video quality measures are useful in rating the performance of a video compression scheme and in providing a certain quality of service (QoS). There are mainly two types of quality measures: subjective and objective.

Subjective measures [17, 50, 51] are the result of human observers providing their opinion of the video quality. These measures are the most reliable quality evaluations since the end user is usually a human viewer. They use *rating scales* such as goodness scales and impairment scales. Goodness scales rate video quality on a scale ranging from excellent to unsatisfactory. Impairment scales rate the video on the basis of the level of degradation in the video when compared with the original or reference video. Although subjective measurements are the most representative video quality measures, they are costly and time consuming.

Objective measures are performed using mathematical equations. A common objective measure is the *peak signal-to-noise ratio* (PSNR) [17] given by

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}[x(m,n)-\tilde{x}(m,n)]^2} = 10\log_{10}\frac{255^2}{MSE} \qquad (2.15)$$

Where $x(m,n)$ and $\tilde{x}(m,n)$ are the original picture and its reconstructed version, respectively, of size $M \times N$.

For video sequences, the PSNR is averaged over the total number of frames. Two important measures for video sequences are the minimum PSNR ($PSNR_{min}$) for all frames and the PSNR variation ($PSNR_{var}$) between the frames. For a sequence of L frames, $PSNR_{min}$ and $PSNR_{var}$ are given by

$$PSNR_{min} = \min\{PSNR(l)\} \qquad \forall\, l \qquad (2.16)$$

and

$$PSNR_{var} = \frac{1}{L}\sum_{l=1}^{L}\left(PSNR(l) - \overline{PSNR}\right)^2 \qquad (2.17)$$

where $\overline{PSNR}$ is the average PSNR over the L frames. An efficient coding scheme should maintain certain minimum picture quality ($PSNR_{min}$) as well as a constant quality (minimum $PSNR_{var}$) throughout the total length of the sequence.

The mean-squared error (MSE) is also a common objective measure and is given by the denominator of Equation 2.15. Another variation on the MSE is the mean-absolute difference (MAD) [16].

Although objective measures like MAD, MSE and PSNR have many practical uses, they may not provide a good measure of the perceptual distortion of images [52]. This is because these measures do not take into account the properties of human perception. Thus, they can not indicate the type of artifacts presented in degraded images, as well as the location of these impairments. Moreover, equal values of MAD, MSE or PSNR for two degraded images do not necessary imply similar subjective quality.

For illustration, Figures 2.4 and 2.5 show the PSNR for two degraded versions of frame No. 11 of Susie sequence. In Figure 2.4, artificial errors have been introduced in two adjacent 8 × 8 blocks on the same line. In Figure 2.5, low pass filtering has been used to remove some of the high spatial frequency content of the picture. Although the PSNR is greater in Figure 2.4, the subjective quality of Figure 2.5 is better.

The drawback in the objective measures together with the complexity and time consumption of the subjective tests let researchers devoting their time to study the human visual system in order to find objective measures of subjective goodness. More details on this can be found in [53, 54, 55, 56, 57].

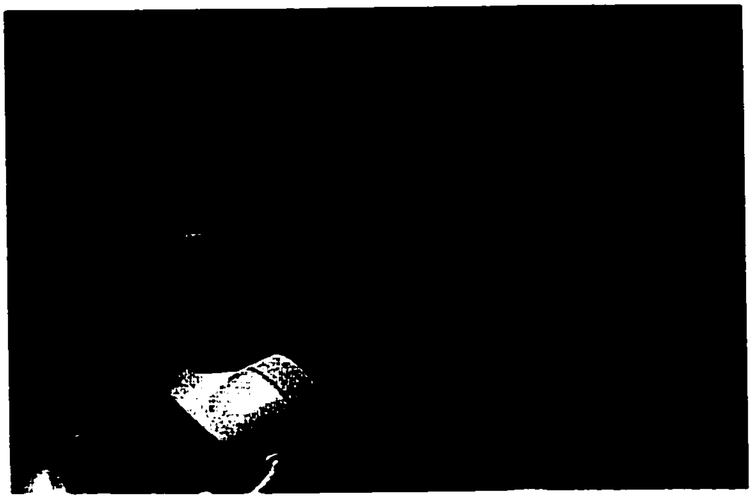


Figure 2.4: Artificial errors introduced. PSNR = 37.72 dB

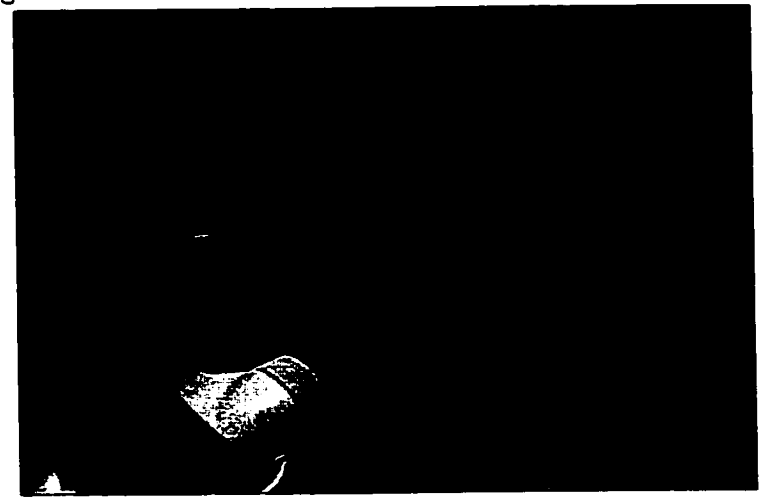


Figure 2.5: Low-pass filtering. PSNR = 35.35 dB

2.4.2 Factors Affecting the Video Quality

There are several important factors that influence the video quality in a compression system. By understanding the impact of each of these, higher video quality can be preserved [58]. Furthermore, a knowledge of these factors allows rating the overall performance of compression systems. Among these factors are the following.

- **Visual complexity:** The number of bits required to compress sequences to a given level of quality varies depending on the visual complexity of the sequences. High detailed sequences with fast motion require more bits than low detailed sequences with moderate motion.

- **Luminance and Chrominance resolutions:** Reducing the resolution of the source video is one of the easiest ways to reduce the data rate in compression. MPEG-1 [20] for example scales images to SIF (Source Input Format) resolution, with luminance resolution of 352 × 240 for NTSC (National Television System Committee) and 352 × 288 for PAL (Phase Alternating Line) and SECAM (Sequentiel Couleur avec Memoire).

  Another common data rate reduction is to subsample the chrominance components even more than the luminance. This takes advantage of the fact that the human visual system is more sensitive to changes in the luminance than changes in the chrominance. For example, MPEG-2 Main profile/Main level (MP@ML) subsample the chrominance by a factor of 2 both in the horizontal and vertical directions with respect to the luminance [21]. This is known as a 4:2:0 video format. While this may be appropriate for home TV viewing, it is inadequate for post-production and editing [58, 59]. Digital video effects and chroma keying require the preservation of more color detail.

- **Bitrate and GOP structure:** In order to estimate the video quality of a compression system, one must know both the bitrate and the frequency of occurrence of the intra-pictures (I) and the non-intra pictures (B and P). In MPEG, a group of picture (GOP) structure commonly defines these frequencies. For example, MPEG-2 MP@ML typically uses a GOP structure of 15 frames as $IBBPBBPBBPBBPBB$. This is commonly referred to by two indices: N, the GOP length, and M, the distance between I/P frames; i.e., $N = 15, M = 3$.

  P and B pictures take less bits to compress than I-pictures. Thus, for a given bitrate longer GOP structures result in higher video quality. Table 2.1 illustrates the relation between the visual complexity of the sequence, the GOP structure, the bitrate (or compression ratio), and the average PSNR. The length of each sequence is 90 frames (each 704 samples/line and 480 lines/frame), with a frame rate of 30 frames/s and a 4:2:2 chroma format. The sequences are coded using the 4:2:2 profile/Main level (4:2:2@ML). The original test sequences are shown in Figure 2.6, Figure 2.7, and Figure 2.8 in order of visual complexity. Table Tennis has the lowest complexity, while Mobile & Calendar has the highest visual complexity. Notice that for a given bitrate and a GOP structure, the PSNR is higher for lower visual complexity sequences. Furthermore, longer GOP structures result, in general, in a higher PSNR.

- **Motion estimation search range:** Motion estimation also influence the quality of the compressed video. A larger search range results in general in a better estimation and a lower residue which will require less bits to compress. Thus for a given bitrate, an efficient motion estimation algorithm will result, in

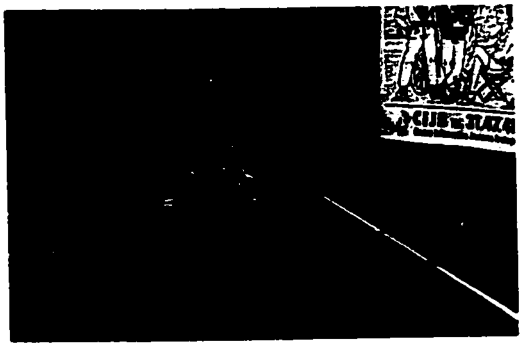


Figure 2.6: Frame No. 30 of Table Tennis

Figure 2.7: Frame No. 15 of Flowers

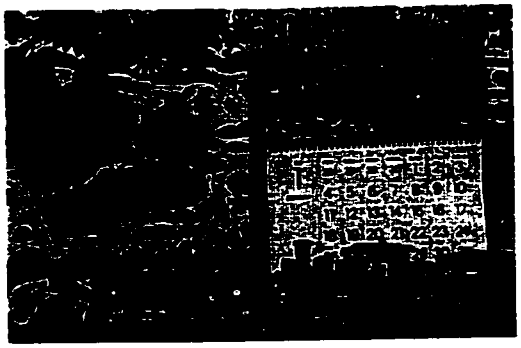


Figure 2.8: Frame No. 15 of Mobile

Table 2.1: PSNR (in dB) of the 4:2:2 profile/Main level for different visual complexity sequences (Y-component) at various bitrates.

Bitrate	Compression factor	GOP structure	PSNR (dB)		
			Tennis	Flowers	Mobile
40 Mb/s	4.06	*I*-only	41.75	40.23	37.66
		IBB	44.57	43.75	40.28
30 Mb/s	5.41	*I*-only	39.11	37.26	33.86
		IBB	42.03	40.94	37.42
20 Mb/s	8.11	*I*-only	35.34	33.63	29.87
		IBB	38.91	37.26	33.98

general, in a higher video quality. However, a larger motion estimation search range requires more computational power, hence, a more complex encoder.

- **Multigeneration:** In some applications, such as a studio environment or an image/video distribution network, it is necessary to compress and decompress images many times. This process is known as *multigeneration* [4, 3, 1, 2]. In a motion-compensated DCT compression scheme like MPEG. multigeneration result in further degradation of the pictures, even though there is no manipulation of the image data and the compression factor is the same in each cycle.

  Shorter GOP structures reduce the degradation in multigeneration. However, higher bitrates are needed to maintain a certain level of video quality. MPEG2 4:2:2 profile/Main level (4:2:2@ML) provides bitrate up to 50 Mb/s which allows the use of short GOPs (e.g. *I*-only, *IB*), thus retaining editing flexibility and quality sufficient for multiple generation of post-production and distribution [21, 60].

  Although longer GOP structure result in higher video quality, they usually suffer more degradation when the video goes through multiple generations.

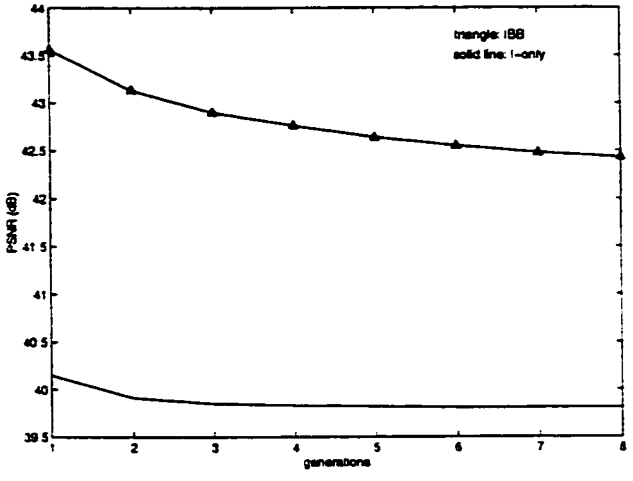


Figure 2.9: Multigeneration of Flowers at 40Mb/s (compression ratio of 4.06) for two different GOP structures.

This is illustrated in Figure 2.9 which shows the relation between the GOP structures and the degradation caused in multigeneration of the video. The PSNR of the sequence Flowers (Y-component) is shown for two GOP structures: I-only, and IBB. The sequence is coded at 40Mb/s, 4:2:2 profile/Main level. Notice that longer GOP structures result, in general, in a higher PSNR at the first generation. However, their degradation through multigeneration is higher.

More details on the degradation mechanisms in multigeneration of MPEG video can be found in [4, 3].

2.4.3 Encoding and Decoding Delays

Most of the computational delays at the encoder depend on the motion compensation algorithms used. On the other hand, an all I-frame coder eliminates the need for motion compensation at the expense of the reduced compression. Large encoding delays may not support real-time applications such as video conferencing.

Decoding delays are also important in real-time video applications. For example, B-frames impose a decoding delay as their prediction depend on future frames. This dependency affects the decoding order of the frames [15]. The decoder must first decode the future frame on which a current B-frame depends on. This also requires additional memory to store the future frame.

In H.261 video compression standard [18] used in video conferencing, B-type macroblocks are not allowed.

2.4.4 Robustness and Error Concealment

Error concealment is an important component of any error robust video codec. The effectiveness of an error concealment strategy highly depends on the concealment method used as well as the ability of the decoder to detect and localize the error, and to resynchronize after an error detection. The goal of error concealment is to recover the missing data at the decoder by utilizing the remaining redundancies in the compressed video bitstream.

In a prioritized Transmission such as ATM networks [24], scalable coding may be used to assist concealment techniques. For example, encoded video data may be partitioned into low-priority data such as high-frequency DCT coefficients, and high-priority data such as addresses, motion vectors and low-frequency DCT coefficients. In the case of network congestion, lower priority ATM cells may be discarded. At the decoder, Lost data can be reconstructed using temporal and spatial interpolation techniques [61, 62].

Error concealment is an area of active research. More details can be found in [61, 62, 63, 64, 65, 66].

2.4.5 Complexity Issues

In addition to providing high perceptual quality, video codecs must also be competitive in terms of implementation and cost. Video compression standards are by nature asymmetric, where the encoder is more complex than the decoder. Further, these standards specify only the decoder and the bitstream, allowing for enhanced encoders as long as they produce a compliant bitstream.

Complexity is reflected by the number of words of memory and combinational logic required to implement the compression algorithm [15]. A common complexity measure is the total number of operations per second and is expressed in *MOPS* (million operations per second). This measure is an important factor in designing of architectures of video processors [16].

Many applications, such as SDTV/HDTV require real-time video compression. The ability of compression algorithms to make significant use of parallelism is an important factor in improving the process speed, especially when the coding algorithms are computationally intensive. Parallel video compression algorithms can be implemented using either hardware or software approaches [67]. Hardware approaches include the design of special architectures to accelerate computations [68], while software approaches may use parallel supercomputers [69, 70].

2.5 Video Compression Impairments

There are several types of picture impairments that may occur in a video compression system. Among these are:

- **Blocking:** is a typical artifact in a block-based DCT compression system which is characterized by the appearance of block edge effects and a visible discontinuity between adjacent blocks, especially in low detail areas of the image [17]. Blocking (or tiling) is usually caused by quantization error in lower DCT coefficients, especially at coarse quantization levels.

Two Approaches to reduce the effect of blocking artifacts are given in [71, 72].

- **Blurring:** is characterized by the loss of sharpness in regions of rapid luminance variation (high detailed regions). It may result when a large percentage of the high-frequency DCT coefficients is set to zero and not transmitted [17].
- **Smearing:** is a localized distortion over a sub-region of the image, characterized by reduced sharpness of edges and spatial detail.
- **Edge busyness:** is characterized by spatial distortion concentrated near the edge of objects.
- **Jerkiness:** is the perception of a nonuniform motion of objects in a scene due to the loss of temporal resolution. That is, smooth and continuous motion is perceived as a series of distinct images.
- **Mosquito noise:** is a distortion characterized by moving artifacts around edges (edge busyness associated with movement).

For illustration, Figure 2.10 shows a zoomed image from frame No. 11 of Susie sequence. Blocking and blurring artifacts for this image are shown in Figures 2.11 and 2.12, respectively.

2.6 Summary of Chapter 2

In this chapter, an overview of different methods used in video compression standards has been presented. The main stages in the coding algorithms of these standards

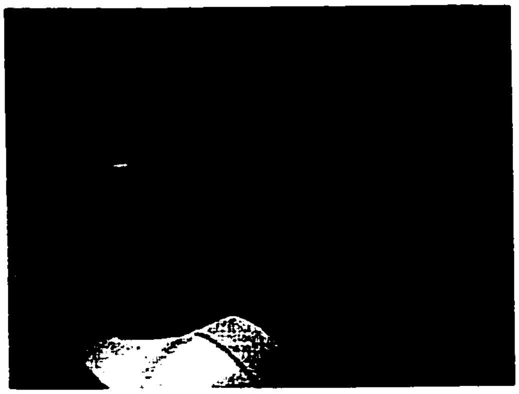


Figure 2.10: Original Image

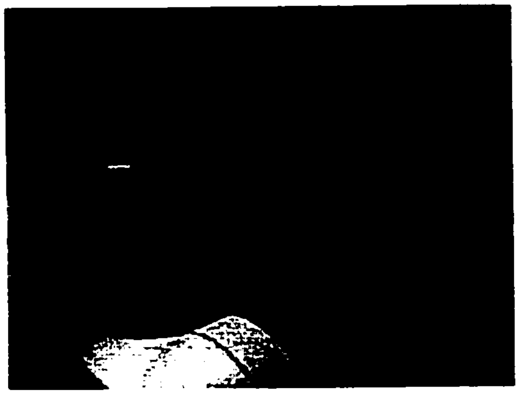


Figure 2.11: Blocking artifact

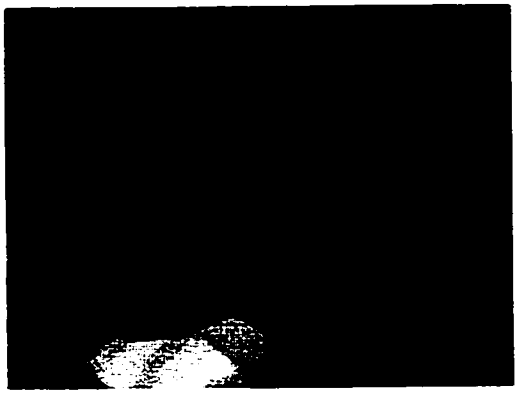


Figure 2.12: Blurring artifact

include an MC-DCT based coding followed by entropy coding of the quantized DCT coefficients. Scalable coding has been also presented as a mean of providing a layered representation of the coded bitstream. This allows interoperability between different services and may assist concealment techniques. In case of network congestion, scalable coding can provide a limited number of bitrates in transmission of MPEG-2 compressed video. However, transcoding is a more reliable approach that provides flexibility in transmission bitrates.

As evaluating the performance of video compression systems is an important factor from a practical viewpoint, it was necessary to discuss various figures of merit including subjective and objective measures, factors affecting the video quality, delays, robustness and error concealment, implementation complexity and cost. The discussion also included different types of picture impairments which may occur in a video compression system.

Chapter 3

Multigeneration of MPEG Compressed Video

3.1 Introduction

In some applications, such as a studio environment or an image/video distribution network, it is necessary to compress and decompress images many times. This process of repeated compression-decompression of images is known as *multigeneration* [3, 4, 1, 2]. In a video postproduction and distribution network, multigeneration maybe unavoidable. In common practice, the video may pass through 3 to 5 generations [58]. Under a compression scheme like MPEG, reconstructed images suffer further degradation at each generation, even though there is no manipulation of the image data and the compression factor is the same in each cycle.

Erdem and Sezan [1] studied the degradation in image quality caused by multiple generations. They used the theory of generalized projections to explain why multigenerations of I, P, and B pictures in MPEG video saturate. Horne et al. [2] have presented multigeneration of MPEG-2 4:2:2 profile for studio environment. They concluded from experiments that the profile is robust in multigeneration, providing a good quality in professional studio and post-production applications. Such applications usually use high bitrates and short group of pictures (GOP) structures (e.g. I-only, IB, etc.) to maintain an image quality sufficient for multiple generations.

In situations where bitrate conversion of MPEG compressed video is necessary, a straightforward approach is to decompress the pre-encoded video then to re-compress it as a second generation video at a lower bitrate using a cascaded decoder/encoder system. This approach however is not an efficient solution in terms of implementation complexity and delay, as the MPEG encoding algorithm is computationally extensive. Moreover, multigeneration errors are introduced from the repeated compression/decompression of MPEG video. Multigeneration errors result from different mechanisms in the MPEG algorithm. However, these mechanisms have not been widely explored. This chapter studies the mechanisms that cause degradation in video quality in a multigeneration environment and quantifies the degradation caused by each mechanism.

A more efficient solution to the problem of bitrate conversion is to transcode the compressed video to the desired bitrate by partially decoding the video then performing the rate conversion by requantizing the DCT coefficients. This is the topic of this thesis and is addressed in detail in subsequent chapters.

3.2 Mechanisms that cause Continuous Degradation in Multigeneration

This chapter presents five mechanisms that contribute to the continued degradation in multigeneration: Pixel Domain Quantization (PDQ), Pixel Domain Clipping (PDC), Compression Control Parameters Variation (CCPV), Motion Vector Re-estimation (MVR) and Error Propagation due to Motion Compensation (EPMC). The chapter also presents other factors that affect the video quality in multigeneration such as group-of-pictures (GOP) structures, changes in picture types between generations, and type or visual complexity of the sequence. Another problem that contributes to the degradation in quality in multigeneration of video is the finite precision of hardware components in real codecs. This precision usually depends on many factors such as registers length, speed, chip area and cost. In a multigeneration environment, finite precision may introduce errors at each generation. However, the study of this problem is beyond the scope of the thesis.

The study of the above degradation mechanisms in multigeneration of MPEG compressed video is a part of the contributions of this thesis [3, 4].

3.2.1 Pixel Domain Quantization (PDQ)

The MPEG compression algorithm uses a DCT coding techniques on 8 × 8 image blocks to explore the spatial correlation between neighboring pixels. To study the effect of Pixel Domain Quantization (round-off) on multigeneration loss, each 8 × 8 block of data is thought of as a vector in R^{64}. The block is transformed into the DCT domain. If finite precision arithmetic effects are neglected, the vector represented by the DCT coefficients and the one represented by the original data are precisely the same, albeit represented in different co-ordinate systems. The DCT coefficients are then independently quantized. In the following, it is assumed a *flat* quantizer (i.e., all coefficients are quantized with the same step size).

Quantization may be thought of as partitioning R^{64} into hypercubes aligned in the direction of the length 64 DCT basis functions. All data vectors that pointed to a point in a particular hypercube are moved to the center of the cube by quantization. In decompression, the quantized DCT coefficients are passed through an inverse DCT. At this point the only degradation is that done by the quantizer in the DCT domain. However, in decompression another quantization takes place, this time in the pixel domain (due to round-off). Normally this quantization is much finer than that in the DCT domain and it is ignored. However in high quality compression the ratios of these two quantizations may become comparable.

In this second quantization the R^{64} space is partitioned into a *second* set of hypercubes, distinct from the first, perhaps by their size but certainly by their orientation. Quantization moves the data vector from where it was after DCT quantization to the center of the particular hypercube it is in, in this second set of hypercubes (see Figure 3.1).

In multigeneration, this second quantization (in the pixel domain) is critical: it moves the data vector again. In the next generation, in the DCT domain the data vector will not start out at the center of one of the DCT hypercubes but will be somewhere else, possibly out of the hypercube that it was at the center of after the last DCT quantization.

The interaction of these two quantizers contributes to the progressive degradation of the image. This degradation will stop when one of the two types of quantization moves the vector but not enough to take it out of the hypercube it was at the center of, in the other domain. Among the two overlaid grids of hypercubes. there will occasionally be pairs (one center from each grid) that lie in each others obliquely aligned hypercubes. When the vector lands on one of these "trapping" pairs it will remain there forever. Until it encounters one of the trapping pairs the block will wander around R^{64}.

Distance Moved by Quantization

To evaluate the distance moved by quantization, consider a 64-dimensional hypercube centered at the origin of size Q in each dimension. Quantization moves the data vector to the center of the hypercube. The maximum distance moved is one from a corner of the hypercube to the reconstruction point (i.e., the origin). Thus, in R^{64}, a flat quantizer with step size Q will move the vector *at most* a distance

$$\sqrt{\sum_{i=1}^{64}(\frac{Q}{2})^2} = 4Q$$

It will take as little as $Q/2$ to move into an adjacent hypercube.

Let Q_1 and Q_2 denote the step sizes of the finer quantizer (i.e., in pixel domain)

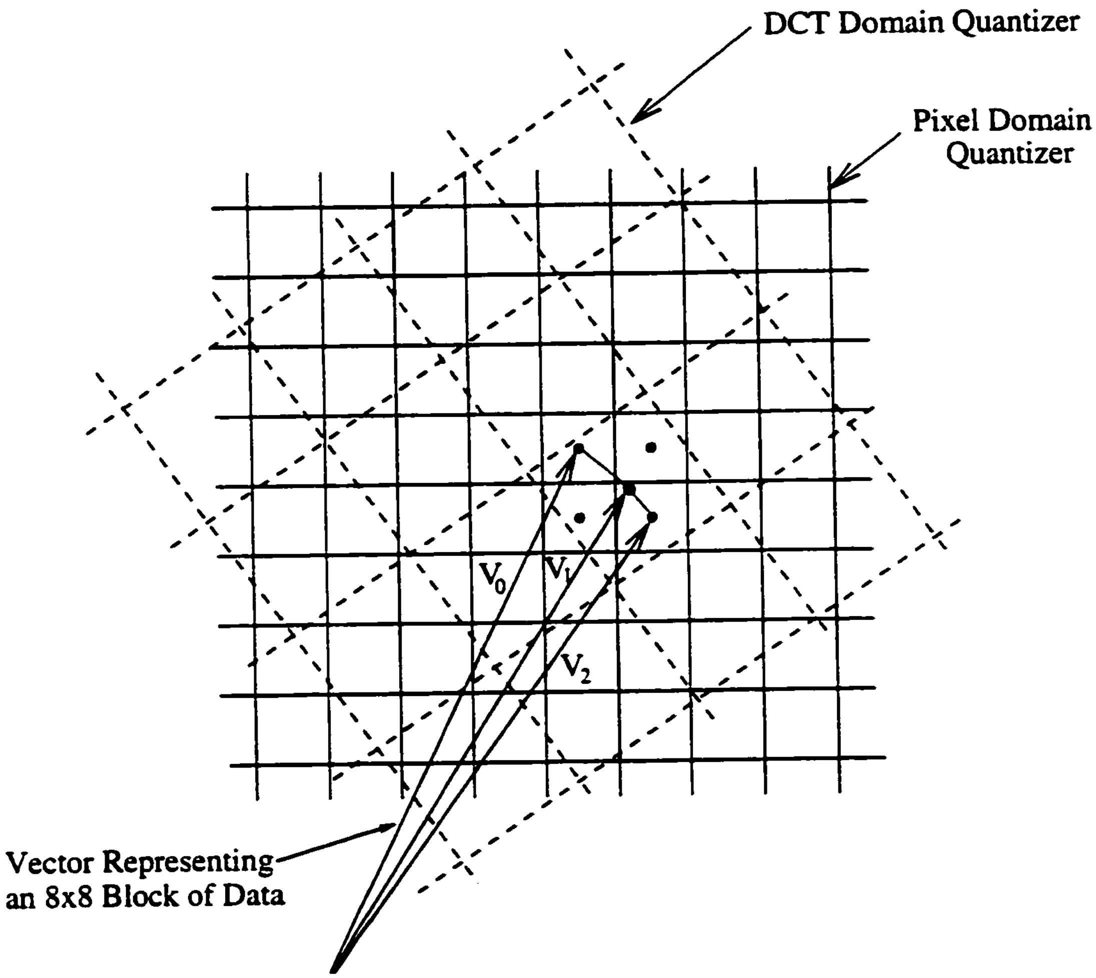


Figure 3.1: Pixel Domain Quantization (PDQ): Two dimensional quantization grids are shown for illustration purposes. V_0: Original data vector in the pixel domain. V_1: Quantized data vector in the DCT domain (compression). V_2: Quantized data vector in the pixel domain (decompression).

and the coarser quantizer (i.e., in DCT domain), respectively. If $(Q_2/2) \geq 4\, Q_1$, i.e., if the coarser quantizer has a step size 8 times or greater than the step size of the finer quantizer, then it will not be possible for the finer quantizer to move the data vector far enough to exit the coarser quantizer's hypercubes. Thus, there will be a trapping pair for every hypercube in the coarser quantization grid.

In order to evaluate the distance moved in the first DCT quantization, consider again a 64-dimensional hypercube centered at the origin of size Q in each dimension. Let $\mathcal{E} = (\mathcal{E}_1,\ \mathcal{E}_2,\ \ldots,\ \mathcal{E}_{64})$ be a random vector, representing the quantization error of the DCT coefficients in each direction, uniformly distributed in this hypercube. Here $\mathcal{E}_1,\ \mathcal{E}_2,\ \ldots,\ \mathcal{E}_{64}$ are i.i.d. with a uniform distribution on $[-Q/2, Q/2]$. If D is the distance of $\mathcal{E}$ from the origin, i.e., the distance moved by quantization, then

$$D^2 = \sum_{i=1}^{64} \mathcal{E}_i^2 \tag{3.1}$$

The Summation of i.i.d. random variables tend to a Gaussian distribution as the number of terms increases (≥ 30). The mean value of D^2 is

$$\mu_{D^2} = E\{D^2\} = \sum_{i=1}^{64} E\{\mathcal{E}_i^2\} \tag{3.2}$$

The value of $E\{\mathcal{E}_i^2\}$ can be evaluated from

$$E\{\mathcal{E}_i^2\} = \int_{\frac{-Q}{2}}^{\frac{Q}{2}} \varepsilon_i^2 f(\varepsilon_i) d\varepsilon_i \tag{3.3}$$

where $f(\varepsilon_i) = \frac{1}{Q}$ for $-\frac{Q}{2} \leq \varepsilon_i \leq \frac{Q}{2}$. This leads to

$$E\{\mathcal{E}_i^2\} = \frac{Q^2}{12} \quad \text{for} \quad i = 1, 2, \ldots, 64 \tag{3.4}$$

Thus, from Equations 3.2 and 3.4 the mean value of D^2 is

$$\mu_{D^2} = \frac{16Q^2}{3} \simeq 5.33Q^2 \tag{3.5}$$

Next, the standard deviation of D^2 is evaluated.

$$Var\{D^2\} = \sum_{i=1}^{64} Var\{\mathcal{E}_i^2\} = \sum_{i=1}^{64}(E\{\mathcal{E}_i^4\} - E^2\{\mathcal{E}_i^2\}) \tag{3.6}$$

The term $E\{\mathcal{E}_i^4\}$ can be evaluated as

$$\begin{aligned} E\{\mathcal{E}_i^4\} &= \int_{\frac{-Q}{2}}^{\frac{Q}{2}} \varepsilon_i^4 f(\varepsilon_i) d\varepsilon_i \\ &= \frac{Q^4}{80} \end{aligned} \tag{3.7}$$

Equations 3.4, 3.6 and 3.7 lead to

$$Var\{D^2\} = \sum_{i=1}^{64}(\frac{Q^4}{80} - \frac{Q^4}{144}) \tag{3.8}$$

yielding

$$Var\{D^2\} = \frac{16Q^4}{45} \tag{3.9}$$

Thus the standard deviation of D^2 is $\sigma_{D^2} = \sqrt{\frac{16Q^4}{45}} \simeq 0.6Q^2$.

Thus, a 95% confidence interval for the value μ_{D^2} gives

$$\begin{aligned} \overline{\mu}_{D^2} &= \mu_{D^2} \pm 1.96\sigma_{D^2} \\ &= (5.33 \pm 1.17)Q^2 \end{aligned} \tag{3.10}$$

Thus, there is a 95% chance that D^2 would lie in the range from $4.2Q^2$ to $6.5Q^2$. Note that D^2 represents the *squared error* introduced by quantization. Clearly from the analysis above, D (Euclidean distance) lies in the range from $2.05Q$ to $2.55Q$ with 95% confidence. This is supported by experiments. Next, the distance moved in second and subsequent quantizations is considered.

The vector representing the original data starts out at the center of a hypercube in the pixel domain. This vector is denoted by V_0. The first quantization moves the vector to the center of a hypercube in the DCT domain. Let this vector be

V_1. Going back to the pixel domain in the next decompression, the vector is moved again (due to PDQ) to the center of a hypercube in the pixel domain. This vector is denoted by V_2 (See Figure 3.2). The amount moved by the first quantization is $D_1 \triangleq ||V_1 - V_0||$ and was evaluated above using statistical arguments. Let the distance moved by the second quantization $D_2 \triangleq ||V_2 - V_1||$. Since quantization selects the nearest reconstruction point, the second quantization moves V_1 to the *closest* hypercube center in the pixel domain. Since it chose V_2 over V_0 it is clear that $D_2 \leq D_1$ (See Figure 3.2). Note that equality holds when $V_0 = V_2$, i.e., In the case of a trapping pair.

Subsequent quantizations will result in non-increasing distortions by similar arguments. Thus PDQ distortion is *non-increasing*. As the distance moved by quantization is getting smaller in each generation, the chances that a vector moves out of the hypercube it was at the center of in the last generation is smaller, thus leading the data vector to a trapping pair. This accelerates the saturation of degradation in multigeneration.

Example 3.2.1.1: Distance Moved by Quantization

In this example, 30 frames (each 704 × 480) from the sequence Table Tennis (Figure 3.3) are used to illustrate the average distance moved by quantization in multigeneration. In each generation, each 8 × 8 block of data is transformed to the DCT domain. The block is then quantized/inverse quantized and transformed back to the pixel domain using an inverse DCT transform. The distance moved by quantization is evaluated for each block through 12 generation. The average distance is then computed by summing all distances and dividing them by the total number of image blocks. Figure 3.4 shows this average distance versus number of generations when both quantizers, in the DCT and pixel domains, are flat and have a quantization step sizes of unity.

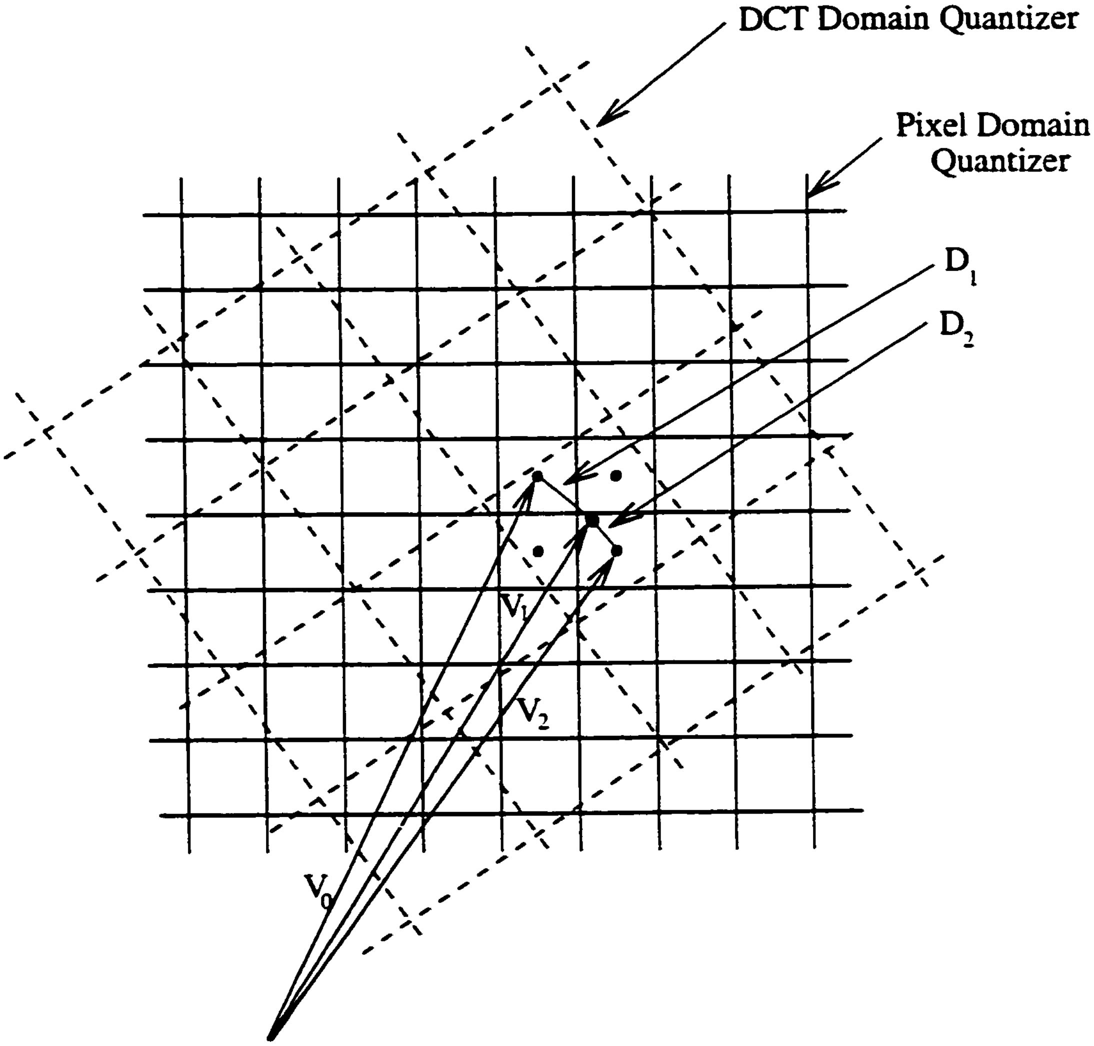


Figure 3.2: Distance moved by quantization. Two dimensional quantization grids are shown for illustration.

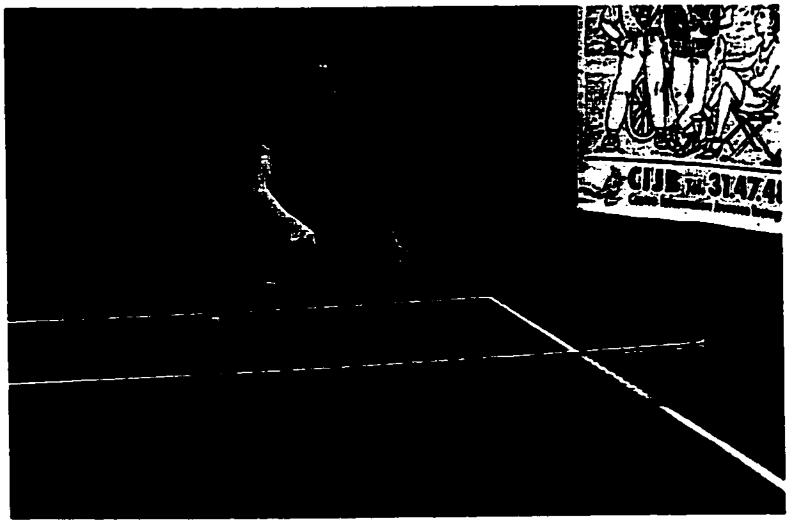


Figure 3.3: Table Tennis sequence.

Note that the average distance moved in the first DCT quantization agrees with the result obtained above using statistical arguments, and that the distance moved by quantization through multigeneration is non-increasing. These experiments have been also repeated using different DCT quantization step sizes and results agree with the above. As the square of the distance represents the squared error introduced by quantization, the non-increasing property of that distance contributes to the saturation of degradation in multigeneration.

Example 3.2.1.2: Profile of how long blocks take to saturate

This example uses 30 frames (each 704 × 480) of the Table Tennis sequence (Figure 3.3) to illustrate the effect of PDQ on multigeneration saturation. Both quantizers, in the DCT and pixel domains, are flat and have $Q = 1$. Each frame is divided into 8 × 8 blocks. Multigeneration is achieved by compressing and decompressing each block until saturation is reached. Figure 3.5 shows the number of blocks (in percentage) and the corresponding number of generations at which multigeneration saturates. All the blocks saturate at less than 13 generations.

In this experiment, multigeneration saturation is slower than in the case where

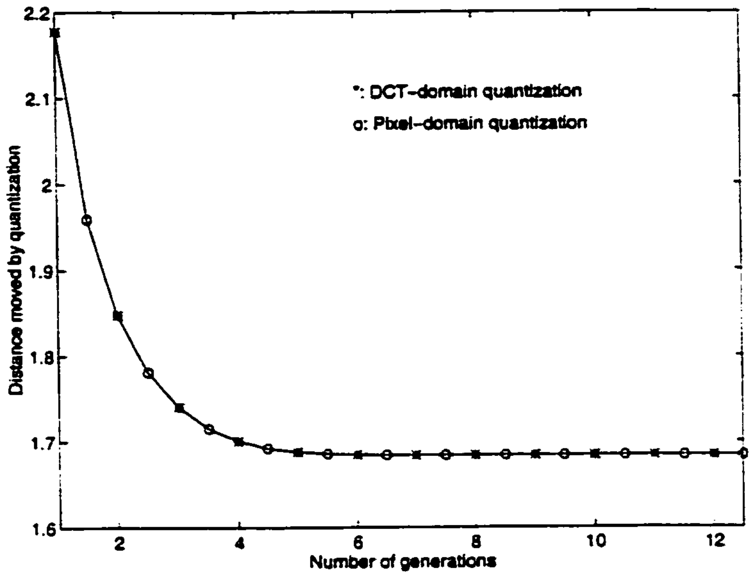


Figure 3.4: Average distance moved by quantization for 12 generations of Table Tennis using a flat-1 DCT quantizer.

the two quantizer step sizes are unequal. This can be seen in Table 3.1 where the same experiment were repeated for a DCT quantizer's step size of 2 and 8. Note that when the ratio of the DCT to the pixel domain quantizers is greater or equal to 8, multigeneration saturates at the second generation. This result is consistent with the previous discussion in this chapter (in this example however, multigeneration also saturates at the second generation for the quantizers ratios 4 and 6). Note that it is not the *size* of the step size of each quantizer which determines the saturation behaviour; but instead the *ratio* of the DCT quantizer to the pixel domain quantizer. The worst case is when the two quantizers have equal size. This is illustrated in Table 3.2 using equal step size of 8 for both quantizers and comparing the results to the case of $Q = 1$ in Table 3.1. The saturation behaviour is similar in both cases.

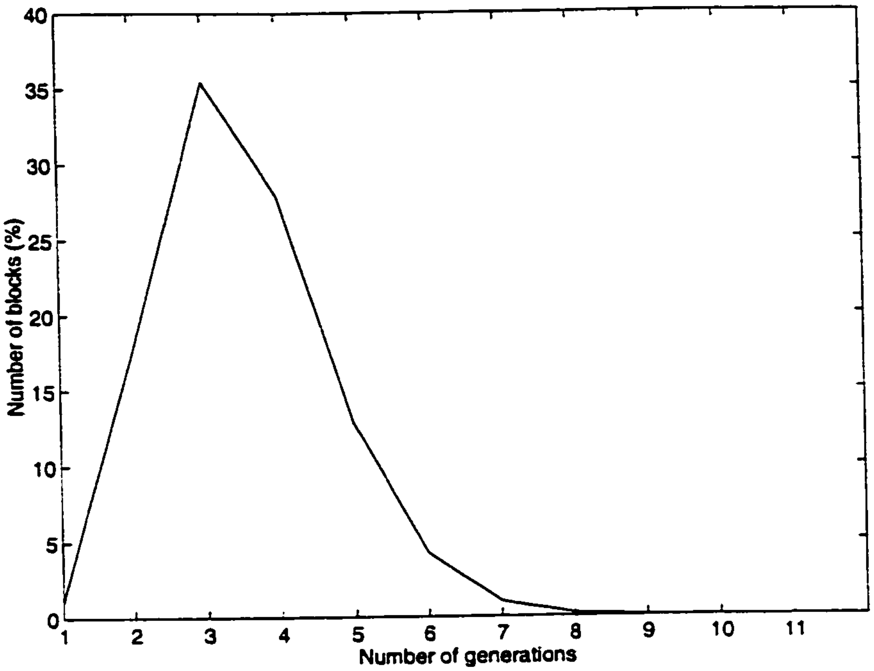


Figure 3.5: PDQ: Percentage of blocks that saturate versus generations for 30 I frames of Table Tennis. Both DCT domain and pixel domain quantizers have $Q = 1$.

Table 3.1: PDQ: Percentage of blocks that saturate versus generations for 30 I frames of Table Tennis, for different DCT quantizer step sizes.

Number of generations	Number of blocks (%)		
	$Q = 1$	$Q = 2$	$Q = 8$
1	1.05	0.001	0
2	17.51	97.172	100
3	35.45	2.817	0
4	27.80	0.01	·
5	12.81	0	·
6	4.21	·	
7	0.98	·	
8	0.17		
9	0.019		
10	0.001		
11	0	0	0
12	0	0	0

Table 3.2: Effect of DCT domain to pixel domain quantizers ratio on multigeneration saturation. For illustration purposes, both DCT and pixel domain quantizers are flat-8. Results are compared to the case of $Q = 1$ in Table 3.1.

Number of generations	1	2	3	4	5	6	7	8	9	10	11	12
Number of blocks (%)	0.18	12.13	34.65	30.928	15.43	5.17	1.24	0.23	0.04	0.002	0	0

3.2.2 Pixel Domain Clipping (PDC)

In practice, data values are clipped in both pixel domain and DCT domain to a specified precision limit. For example, in MPEG-2 standard [21], data are clipped to the range $[-2048 : +2047]$ in the DCT domain, and to the range $[0 : +255]$ in the pixel domain. To study the effect of PDC on multigeneration, PDQ is ignored in the analysis.

Consider the quantization of the DCT coefficients. The quantizer step size of the (i, j)th DCT coefficient includes two terms: a quantization coefficient $q(i, j)$ (from the quantization matrix) which defines a minimum step size for the particular DCT coefficient, and a quantization scaling parameter $qscale$ used to control the output bitrate. Let V be a 64-dimensional vector representing the reconstructed data in the pixel domain at the current generation, and let V^c be the data values after clipping. Let $B_{i,j}$ be the (i, j)th DCT basis vector. Define the clipping error as the difference $V - V^c$. Thus, the term $B_{i,j}^T \cdot (V - V^c)$ represents the change in the (i, j)th DCT coefficient, at the next generation, due to clipping in the pixel domain at the current generation. Let $qscale \times q(i, j)$ be the quantizer step size for the (i, j)th DCT coefficient. Thus, if

$$\left|B_{i,j}^T (V - V^c)\right| > qscale\frac{q(i,j)}{2} \tag{3.11}$$

then in the next generation the quantized DCT coefficient will move (depending on the value of the left term in equation 3.11) an integral multiple of the step size $qscale \times q(i,j)$.

Example 3.2.2.1: Pixel Domain Clipping

Figure 3.6: Original Lena.

Figure 3.7: Original Peppers.

The effect of PDC on the DC coefficient is illustrated in this example. The test image Lena shown in Figure 3.6 is used. Multigeneration of an 8 × 8 data block of Lena (line 321 to 328, samples 177 to 184) is presented in Figures 3.8 and 3.9 for the first and second generations, respectively. The DCT coefficients of the data block are quantized with the default intra quantization matrix specified in MPEG2 video standard [21] and with $qscale = 2$. In Reconstruction, an inverse DCT is performed followed by a quantization (rounding) in the pixel domain. The reconstructed block contains negative values which are then clipped to zero. PDC error in the first generation is shown in Figure 3.8.(d). Note that the sum of the clipping error equals −73, and with $q(0,0) = 8$ and $qscale = 2$ for this example, the condition in Equation 3.11 is satisfied for $i, j = 0$. Thus in the next generation, the quantized DC coefficient will move an amount that is integral multiple of 16. In this example, as $\frac{-73}{8} = -9.125$, the DC coefficient will move to the adjacent higher quantization cell.

The generation error due to PDC is evaluated by subtracting the reconstructed data block in the second generation from its corresponding block in the first generation. This is shown in Figure 3.9. Note that the data values in Figure 3.9.(b). are shifted from their corresponding values in Figure 3.8.(c). by a constant value of 2 as only the DC coefficient has changed between the two generations. As the sum of clipping error in the second generation (Figure 3.9.(c).) is -54, Equation 3.11 does not hold and there will be no further changes in the value of the DC coefficient. In fact, the block saturates after the second generation.

Pixel Domain Clipping: Example 3.2.2.2

An 8×8 data block (line 169 to 176, samples 49 to 56) of Peppers image (Figure 3.7) is used, with flat-16 quantization matrix and $qscale = 1$. In the first generation, the left-hand-side of Equation 3.11 equals 8.86 for $i = 0, j = 1$, and the inequality is satisfied. In the next generation, the corresponding lower-frequency AC coefficient moves to the next quantization cell. Further, Equation 3.11 is satisfied for the DC coefficient. This results in a generation error shown in Figure 3.10.(b). The loss in PSNR for the 8×8 block is evaluated as $PSNR^{(1)} - PSNR^{(2)}$ where the subscripts 1 and 2 denote first and second generations. This loss is 2.6 dB. Here, the PSNR is evaluated over the 8×8 block and not over the entire image. Note that in Examples 3.2.2.a. and 3.2.2.b. the DCT domain to pixel domain quantizers ratios are greater than 8. Hence PDQ does not contribute to the generation loss.

Clipping has a tendency to affect lower-frequency DCT coefficients. Clipping error is impulsive, meaning that the error energy is spread out among all frequencies. However, when multiple pixels are clipped in the same block, they are usually all clipped in the same direction. Thus clipping error will have a DC bias. This coupled with the smaller quantization step sizes of low frequency coefficients, results in most

23	16	6	3	3	8	76	160
17	15	7	8	4	15	128	163
13	6	5	4	5	33	154	169
14	12	8	5	10	65	166	165
9	9	4	8	24	113	152	150
10	10	6	11	56	135	132	139
21	21	31	44	84	121	113	126
56	51	66	73	88	85	97	115

(a) Data block samples

448	−352	190	−44	0	0	0	0
−96	−32	88	−96	0	0	0	0
0	88	−52	−54	58	0	0	0
0	0	0	54	0	0	0	0
0	52	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) Quantized DCT coefficients

22	19	23	19	−1	4	74	156
−9	5	16	9	−4	24	107	187
−17	6	13	−4	−4	49	134	196
10	20	9	−13	8	77	142	169
32	22	−3	−12	35	109	148	144
32	12	−6	9	65	125	147	140
36	21	21	48	81	103	119	131
51	44	57	82	83	69	82	113

(c) IDCT + Quantization

0	0	0	0	−1	0	0	0
−9	0	0	0	−4	0	0	0
−17	0	0	−4	−4	0	0	0
0	0	0	−13	0	0	0	0
0	0	−3	−12	0	0	0	0
0	0	−6	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(d) Clipping error (Clipping to [0 − 255])

Figure 3.8: PDC: First generation of an 8 × 8 data block of Lena (no PDQ effect).

464	−352	190	−44	0	0	0	0
−96	−32	88	−96	0	0	0	0
0	88	−52	−54	58	0	0	0
0	0	0	54	0	0	0	0
0	52	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) Quantized DCT coefficients

24	21	25	21	1	6	76	158
−7	7	18	11	−2	26	109	189
−15	8	15	−2	−2	51	136	198
12	22	11	−11	10	79	144	171
34	24	−1	−10	37	111	150	146
34	14	−4	11	67	127	149	142
38	23	23	50	83	105	121	133
53	46	59	84	85	71	84	115

(b) IDCT + Quantization

0	0	0	0	0	0	0	0
−7	0	0	0	−2	0	0	0
−15	0	0	−2	−2	0	0	0
0	0	0	−11	0	0	0	0
0	0	−1	−10	0	0	0	0
0	0	−4	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c) Clipping error (Clipping to [0 − 255])

−2	−2	−2	−2	−1	−2	−2	−2
0	−2	−2	−2	0	−2	−2	−2
0	−2	−2	0	0	−2	−2	−2
−2	−2	−2	0	−2	−2	−2	−2
−2	−2	0	0	−2	−2	−2	−2
−2	−2	0	−2	−2	−2	−2	−2
−2	−2	−2	−2	−2	−2	−2	−2
−2	−2	−2	−2	−2	−2	−2	−2

(d) Generation error

Figure 3.9: PDC: Second generation of the 8 × 8 data block of Lena.

$$\begin{array}{rrrrrrrr}
5 & 0 & 0 & 0 & 0 & 0 & 50 & 132 \\
8 & 0 & 0 & 0 & 0 & 0 & 42 & 110 \\
2 & 0 & 0 & 0 & 0 & 0 & 36 & 107 \\
0 & 0 & 0 & 0 & 0 & 3 & 37 & 120 \\
0 & 0 & 0 & 0 & 0 & 0 & 26 & 102 \\
0 & 0 & 0 & 0 & 0 & 0 & 34 & 87 \\
0 & 0 & 0 & 0 & 0 & 0 & 10 & 69 \\
0 & 0 & 0 & 0 & 0 & 0 & 16 & 54
\end{array}$$

(a) Data block samples

$$\begin{array}{rrrrrrrr}
-5 & 0 & 0 & 0 & -2 & 0 & 1 & 1 \\
-5 & -4 & -2 & -2 & -1 & -1 & 1 & 1 \\
-5 & -4 & -3 & -3 & -1 & 0 & 0 & 1 \\
0 & 0 & -3 & -2 & -1 & 0 & 0 & 1 \\
0 & 0 & -2 & -1 & -2 & -1 & 1 & 1 \\
-1 & -2 & -3 & 0 & -1 & -1 & 1 & 1 \\
-5 & -4 & -4 & 0 & -1 & 0 & 0 & 1 \\
-5 & -4 & -3 & 0 & -1 & -1 & 1 & 1
\end{array}$$

(b) Generation error

Figure 3.10: PDC: Generation error of an 8 × 8 data block of Peppers (no PDQ effect).

clipping error affecting only the DC coefficient.

3.2.3 Compression Control Parameters Variation (CCPV)

In multigeneration, rate control algorithms may use different quantization scaling parameters at subsequent generations (due to a change in the picture activity) for the same data block. A change in qscale can be thought of as resizing the 64-dimensional hypercube of the DCT quantizer. Thus data vectors which already were in trapping pairs at a given generation may escape from their trap, hence slowing down multigeneration saturation and causing more degradation in the picture quality. This is illustrated in Section 3.3.

3.2.4 Motion Vector Re-estimation (MVR)

If motion vectors are re-estimated at each generation, they may differ from the motion vectors used in the first generation. Different motion vectors will result in different reference blocks for predicted data blocks. The re-estimated motion vectors may be optimum in some sense (for instance they may minimize the size of the residue as measured by absolute error for the current generation); however. the first generation motion vectors probably more closely represent the *true* motion of the objects in the video. Moreover, minimizing the residue may not always lead to fewer bits in picture coding if the residue contains high frequency spikes due to non-smoothness of motion vectors.

Experiments show that using motion vectors of the first generation reduces multigeneration loss. This requires that picture types as well as macroblock types not be altered between generations. Although using the same motion vectors requires storing them for subsequent generations, it has the advantage of reducing the complexity at the encoder, as well as the generation loss. However, motion vectors may need to be updated when multigeneration involves production manipulations between generations. Erdem and Sezan also suggested that motion vectors should not be altered from generation to generation [1].

3.2.5 Error Propagation due to Motion Compensation (EPMC)

Consider a sequence represented as

$$I_0B_1B_2P_3B_4B_5P_6B_7B_8I_9\ldots$$

In this section, the predicted frame P_3 will be examined through two generations. Here subscripts refer to the frame number, superscripts refer to the generation, hats refer to estimated frames, and tildas refer to reconstructed quantities. Motion estimation of P_3 from the reference frame I_0 results in motion vectors $MV_{30}^{(1)}$. Let $\tilde{I}_0^{(1)}$ be the reconstructed frame of I_0 at the first generation. Using $MV_{30}^{(1)}$ and $\tilde{I}_0^{(1)}$, an estimate $\hat{P}_3^{(1)}$ of frame P_3 is formed. The residue $R_3^{(1)}$ resulting from Motion Compensated Differential Pulse Code Modulation (MC-DPCM) is

$$R_3^{(1)} = P_3 - \hat{P}_3^{(1)} \tag{3.12}$$

$R_3^{(1)}$ is then transformed (DCT) and quantized. Let $\tilde{R}_3^{(1)}$ be the reconstruction of $R_3^{(1)}$ at the decoder. The decoder will reconstruct $P_3^{(1)}$ as

$$\tilde{P}_3^{(1)} = \tilde{R}_3^{(1)} + \hat{P}_3^{(1)} \tag{3.13}$$

In the second generation, $\tilde{P}_3^{(1)}$ is predicted from $\tilde{I}_0^{(2)}$, the reconstructed version of the reference frame $\tilde{I}_0^{(1)}$. If $\tilde{I}_0^{(2)}$is different than $\tilde{I}_0^{(1)}$ due to PDQ, PDC or CCPV; or if $MV_{30}^{(2)}$ is different from $MV_{30}^{(1)}$ because of motion vector re-estimation (MVR);

or if different modes are chosen: then the estimated frame $\hat{P}_3^{(2)}$ will be different from $\hat{P}_3^{(1)}$. This will result in different residues $R_3^{(2)}$

$$R_3^{(2)} = \tilde{P}_3^{(1)} - \hat{P}_3^{(2)} \tag{3.14}$$

Consequently, the reconstructed frame $\tilde{P}_3^{(2)}$ at the second generation will be different from the reconstructed frame $\tilde{P}_3^{(1)}$ at the first generation, even if no PDQ, PDC or CCPV error is introduced in the transform coding of the residue $R_3^{(2)}$. $\tilde{P}_3^{(2)}$ is given by

$$\tilde{P}_3^{(2)} = \tilde{R}_3^{(2)} + \hat{P}_3^{(2)} \tag{3.15}$$

Moreover, even if $\tilde{I}_0^{(2)}$ is identical to $\tilde{I}_0^{(1)}$, and the motion vectors $\vec{MV}_{30}^{(1)}$ and $\vec{MV}_{30}^{(2)}$ are the same, the residue $R_3^{(2)}$ will still be different from $R_3^{(1)}$, as the frame P_3 is not available at the second generation, but only its reconstructed version $\tilde{P}_3^{(1)}$ (refer to Equation 3.12 and 3.14). This will yield a reconstructed frame $\tilde{P}_3^{(2)}$ (Equation 3.15) different from $\tilde{P}_3^{(1)}$. This problem recurs in subsequent generations. Note that this error propagation between predicted frames is more of a problem for higher bitrates as the DCT quantizer is finer, possibly leading the residue data vector to a different quantization hypercube in the next generation, in the DCT domain. Similar analysis can be done for B frames.

Thus, the above concludes that in multigeneration, EPMC propagates the degradation, caused by other mechanisms, from reference frames to frames that are predicted from them. Furthermore, even if there was no degradation in reference frames, EPMC still introduces errors at each generation. This is because a predicted frame at a current generation is a reconstruction version of a corresponding predicted frame from the previous generation.

3.3 Experiments

The test sequences used in this section are Table Tennis and Flowers. The sequence consists of 60 frames with a frame rate of 30 frames/s and a color sampling ratio of 4:2:0. The coding bitrates are 5, 10 and 15 Mb/s. The number of frames in a group of pictures (GOP) is 15, with a structure IBBPBBPBBPBBPBB. The default intra and non-intra quantization matrices specified in MPEG-2 Video standard [21, 39] are used. The PSNR is used as an objective measure to quantify the amount of degradation caused by each mechanism.

The strategy used in the following experiment is to "turn off" each degradation mechanism and evaluate its effect. Thus, to evaluate the effect of PDQ on multigeneration degradation, the quantization in the pixel domain is turned off and the average PSNR is computed for 20 generations. For PDC, values less than 0 or greater than 255 in the pixel domain are allowed. CCPV and MVR mechanisms are evaluated by using motion vectors and quantization scaling parameters of the first generation for each subsequent generation. As for EPMC, it is tested by turning off PDQ, PDC, CCPV, and MVR mechanisms.

The organization of the subsequent sections is as follows. Section 3.3.1 test PDQ mechanism. Section 3.3.2 illustrates MVR mechanism. The effect of both CCPV and MVR mechanisms on multigeneration loss is tested in Section 3.3.3. Next, EPMC is illustrated in Section 3.3.4. The simultaneous effect of PDQ, PDC, CCPV, MVR, and EPMC mechanisms is provided in Section 3.3.5.

Other experiments that address different factors affecting multigeneration loss are presented in Section 3.3.6. This section provides results that were presented in a technical report to the Canadian Broadcasting Corporation (CBC) which was supporting me during my NSERC Industrial Postgraduate Scholarship program.

3.3.1 PDQ Mechanism

Figure 3.11 shows the average PSNR of Table Tennis at bitrates 15 and 5 Mb/s for 20 generations. In (a) and (c) all the mechanisms that contribute to the continued degradation in multigeneration are included, while (b) and (d) show the PSNR with no PDQ. Notice that multigeneration average PSNR increases when PDQ mechanism is disabled. The difference in PSNR after 20 generations is 6.77 dB at 15 Mb/s and 0.66 dB at 5 Mb/s. Further, this difference is greater for higher bitrates (lower compression ratios) as the ratio between the DCT domain and the pixel domain quantizers becomes comparable, leading to more interaction between them, and hence further degradation at each generation. This result agrees with Section 3.2.1. Notice the crossover of the 15Mb/s and 5Mb/s curves where all degradation mechanisms are included (curves (a) and (c)). While the bit budget spent on the coded video of curve (a) is 3 times higher than the bit budget spent on the video of curve (c), the quality of both sequences after 20 generation is very close due to multigeneration loss.

3.3.2 MVR Mechanism

Figure 3.12 shows multigeneration average PSNR of Table Tennis at bitrates 15 and 5 Mb/s. In (a) and (c) all mechanisms that cause degradation in the picture quality are included, while (b) and (d) use the motion vectors of the first generation for each subsequent generation. Motion vector re-estimation (cases (a) and (c)) results in a PSNR loss of 0.77 dB and 0.42 dB at the 20th generation, for the bitrates 15 and 5 Mb/s, respectively.

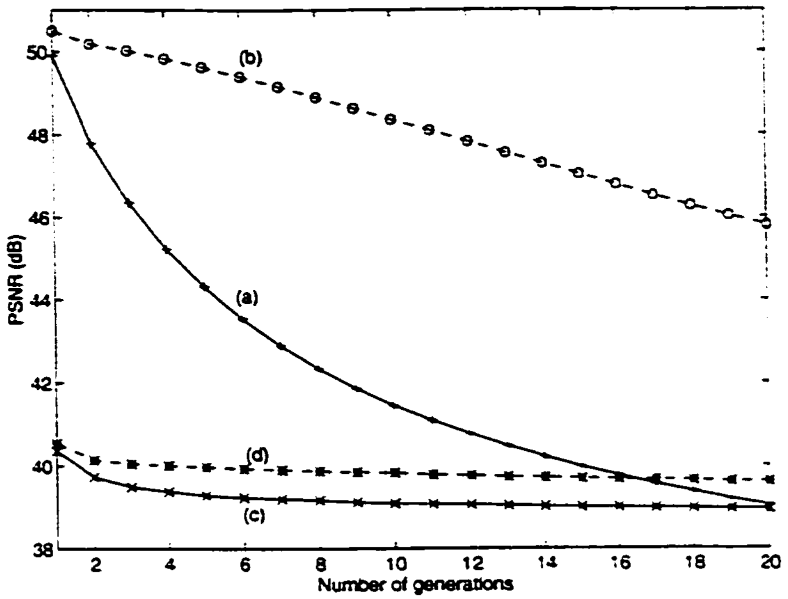


Figure 3.11: Multigeneration of Table Tennis (Y-component). (a) 15 Mb/s. all mechanisms. (b) 15 Mb/s. no PDQ. (c) 5 Mb/s. all mechanisms. (d) 5 Mb/s. no PDQ.

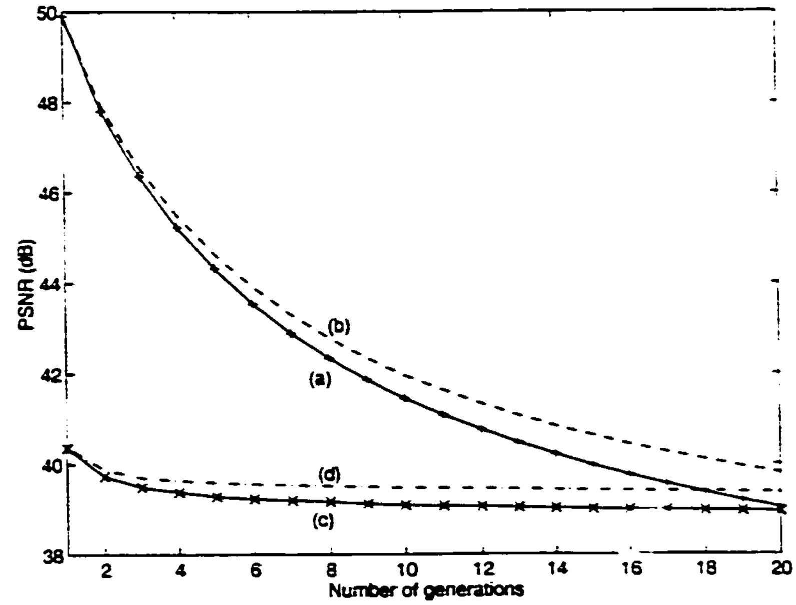


Figure 3.12: Multigeneration of Table Tennis (Y-component). (a) 15 Mb/s, all mechanisms. (b) 15 Mb/s, no MVR. (c) 5 Mb/s, all mechanisms. (d) 5 Mb/s, no MVR.

3.3.3 CCPV and MVR Mechanisms

In this experiment, CCPV and MVR mechanisms are combined. The average PSNR is shown in Figure 3.13 for 20 generations of Table Tennis at 5 Mb/s. In (a) all mechanisms contributing to multigeneration loss are included. Case (b) uses motion vectors of the first generation for subsequent generations. In (c) motion vectors and quantization scaling parameters of the first generation are used through multigeneration. It was observed that using the quantization scaling parameters of the first generation (case (c)) results in an increase in the average PSNR of lower bitrate video (compression ratio $\geq$ 6:1) while maintaining the target bitrate relatively constant.

Because the amount of degradation in Figure 3.13.(c). is small, this suggests that CCPV and MVR are the most important mechanisms for the above video compression ratios.

The above experiment were also repeated at 15 Mb/s, however the coding bitrate could not be controlled through multigeneration. as using the quantization scaling parameters of the first generation, the rate control algorithm is disabled at subsequent generations.

3.3.4 EPMC Mechanism

Figure 3.14 show multigeneration average PSNR of Table Tennis at 15 and 5 Mb/s. Curves (a) and (c) show the PSNR when all mechanisms contributing to multigeneration degradation are included, while (b) and (d) show the PSNR when PDQ, PDC, CCPV and MVR mechanisms are disabled (i.e., only EPMC mechanism is in effect).

Although CCPV mechanism is disabled in (b) and (d), the coding bitrates

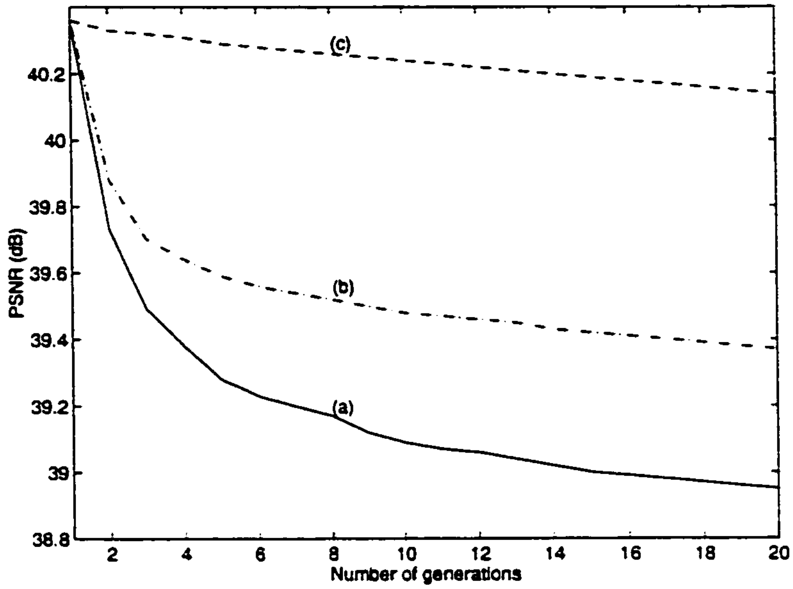


Figure 3.13: Multigeneration of Table Tennis at 5Mb/s (Y-component). (a) All mechanisms. (b) No MVR. (c) No MVR, qscale of the first generation.

is found constant through multigeneration. Note that the loss in PSNR after 20 generations (i.e., $PSNR^{(1)} - PSNR^{(20)}$) due to EPMC is 4.5 dB and 0.1 dB for the bitrates 15 and 5 Mb/s, respectively. This leads us to conclude that EPMC results in more degradation for higher bitrates. This result is consistent with the discussion in Section 3.2.5, that is, the degradation caused by EPMC when all other mechanisms are disabled (curves (b) and (d)) results from the fact that the predicted frames at a current generation is the reconstruction versions of their corresponding predicted frames from the previous generation. Moreover, For the curves (a) and (c) in Figure 3.14, All mechanisms, including EPMC, contribute to the multigeneration loss. In this case, in addition to the degradation mentioned above, EPMC also propagates the errors from reference frames to the frames that are predicted from them, thus, acting as an amplifier.

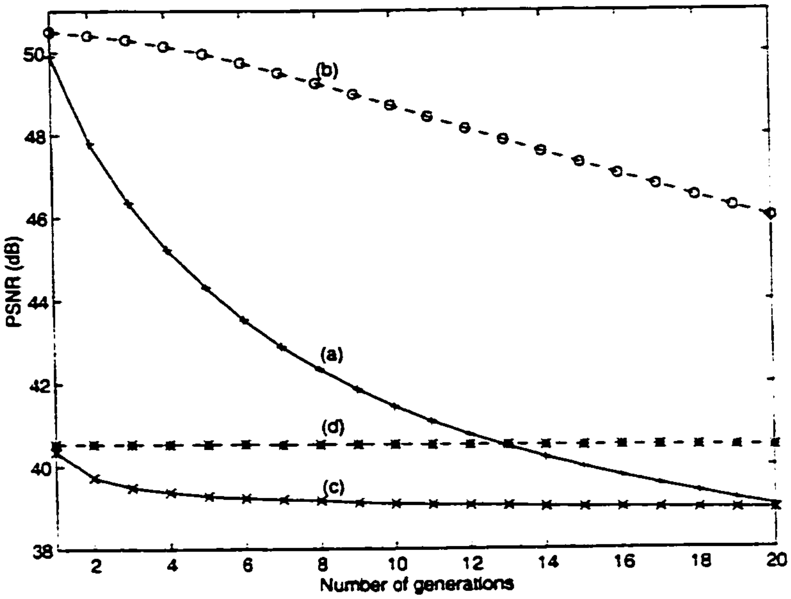


Figure 3.14: Multigeneration of Table Tennis (Y-component). (a) 15 Mb/s. all mechanisms. (b) 15 Mb/s, EPMC only. (c) 5 Mb/s. all mechanisms. (d) 5 Mb/s. EPMC only.

3.3.5 Simultaneous Effect of All Mechanisms

In this experiment, Table Tennis and Flowers sequences are used. The sequence were compressed and decompressed through 20 generations at bitrates (R) of 15, 10 and 5 Mb/s.

Tables 3.3 and 3.4 show the average PSNR when all mechanisms that contribute to multigeneration degradation are included. Note that the loss after 20 generations (defined as $L^{(20)} = PSNR^{(1)} - PSNR^{(20)}$) increases in general with bitrates, that is, higher bit-rates video suffer more degradation through multigeneration.

Although the difference in video quality (measured in PSNR) of the luminance component (Y-component) in Table 3.3 is significant at the first generation for the above bitrates, the quality at the 20th generation is comparable. Further, the PSNR of the chrominance components (Cb and Cr components) in Table 3.3 after 20 generations at 5 Mb/s is higher than the corresponding PSNR at 10 and 15 Mb/s. This behavior however is image dependent and differs for different sequences.

3.3.6 Other Factors Affecting Multigeneration Loss

This section addresses factors affecting the video quality in multigeneration of the MPEG-2 4:2:2 profile. This profile is used in the professional studio and post production environment, and in distribution between TV networks and their affiliates [2, 59].

In the experiments, parameters such as bitrate, GOP structure, quantization matrices and picture types (I, P, or B) are changed, and the multigeneration average peak signal-to-noise ratio (PSNR) is evaluated for 8 generations. Multigeneration characteristics for different visual complexity sequences were also addressed.

GOP Structures and Multigeneration

The shorter the GOP structure is, the lower are the errors introduced by multiple generation of the video. The degradation mechanisms that are affected most by the GOP structures are EPMC (Error Propagation due to Motion Compensation) and MVR (Motion Vector Re-estimation) [4, 3].

In practice, a nonlinear editing system may use MPEG-2 4:2:2 profile with only Intra coded pictures at bitrates 20 to 40 Mb/s [58]. Using only I-frames avoids some of the mechanisms that contribute to the degradation in video quality during multiple generations. Specifically, EPMC and MVR mechanisms. EPMC mechanism dominates the multigeneration loss for higher bitrate video as previously discussed in this chapter. I-frame structure also avoids changes in picture types which results whenever a temporal shift occurs between two generations.

Table 3.3: Effect of all mechanisms on multigeneration average PSNR (in dB) of Table Tennis. The term R denotes the bitrate, and $L^{(20)} = PSNR^{(1)} - PSNR^{(20)}$.

Y-component						
R	$PSNR^{(1)}$	$PSNR^{(2)}$	$PSNR^{(3)}$	$PSNR^{(4)}$	$PSNR^{(20)}$	$L^{(20)}$
15	49.91	47.80	46.36	45.23	39.03	10.88
10	46.00	44.53	43.58	42.83	39.03	6.97
5	40.36	39.73	39.49	39.38	38.95	1.41
Cb-component						
R	$PSNR^{(1)}$	$PSNR^{(2)}$	$PSNR^{(3)}$	$PSNR^{(4)}$	$PSNR^{(20)}$	$L^{(20)}$
15	50.33	48.68	47.53	46.60	41.11	9.22
10	46.98	45.82	45.05	44.42	40.71	6.27
5	43.39	43.06	42.93	42.87	42.61	0.78
Cr-component						
R	$PSNR^{(1)}$	$PSNR^{(2)}$	$PSNR^{(3)}$	$PSNR^{(4)}$	$PSNR^{(20)}$	$L^{(20)}$
15	50.63	49.00	47.86	46.95	41.37	9.26
10	47.36	46.17	45.39	44.76	41.00	6.36
5	43.66	43.27	43.12	43.05	42.76	0.90

Table 3.4: Effect of all mechanisms on multigeneration average PSNR (in dB) of Flowers. The term R denotes the bitrate, and $L^{(20)} = PSNR^{(1)} - PSNR^{(20)}$.

Y-component						
R	$PSNR^{(1)}$	$PSNR^{(2)}$	$PSNR^{(3)}$	$PSNR^{(4)}$	$PSNR^{(20)}$	$L^{(20)}$
15	45.87	44.66	43.83	43.19	38.26	7.61
10	41.41	40.66	40.29	40.02	37.83	3.58
5	34.92	34.59	34.51	34.48	34.35	0.57
Cb-component						
R	$PSNR^{(1)}$	$PSNR^{(2)}$	$PSNR^{(3)}$	$PSNR^{(4)}$	$PSNR^{(20)}$	$L^{(20)}$
15	45.69	44.76	44.12	43.66	39.58	6.11
10	41.61	41.01	40.74	40.56	38.86	2.75
5	36.25	36.07	36.04	36.03	35.94	0.31
Cr-component						
R	$PSNR^{(1)}$	$PSNR^{(2)}$	$PSNR^{(3)}$	$PSNR^{(4)}$	$PSNR^{(20)}$	$L^{(20)}$
15	45.63	44.75	44.14	43.65	39.60	6.03
10	41.80	41.30	41.06	40.89	39.32	2.48
5	37.05	36.93	36.92	36.91	36.85	0.20

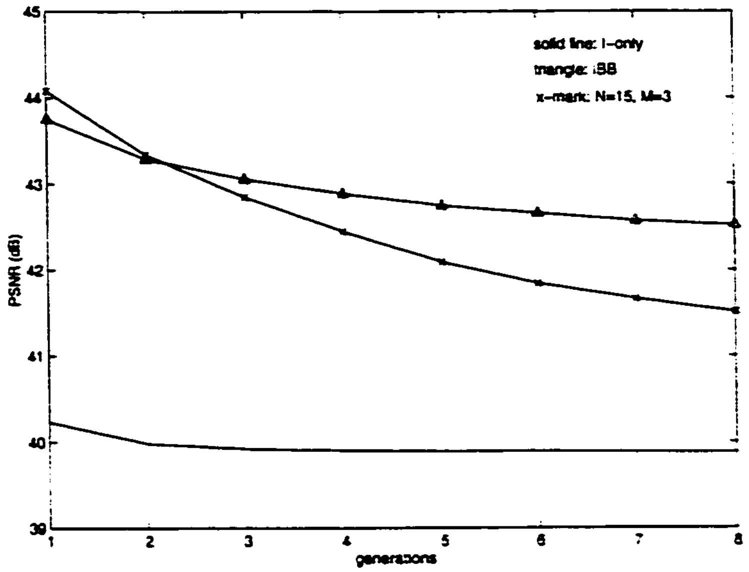


Figure 3.15: Multigeneration of the MPEG-2 4:2:2 profile for different GOP structures. Flowers at 40Mb/s (Y-component).

Figure 3.15 shows multigeneration average PSNR of Flowers sequence at 40Mb/s Notice that longer GOP structures result. in general. in a higher PSNR at the first generation. However, multigeneration loss is higher due to EPMC mechanism.

Temporal Shifts

Temporal shifts is the change in picture types between generations. this may occur for example after a video editing. In the experiments, temporal shifts are performed after the first and the fourth generations, using a one-frame shift.

Figure 3.16 shows the effect of temporal shifts on multigeneration average PSNR (Y-component) of Table Tennis for GOP structures IBB and N=15. M=3 at bitrate of 30Mb/s. Notice that temporal shifts result in more degradation in multigeneration of the video.

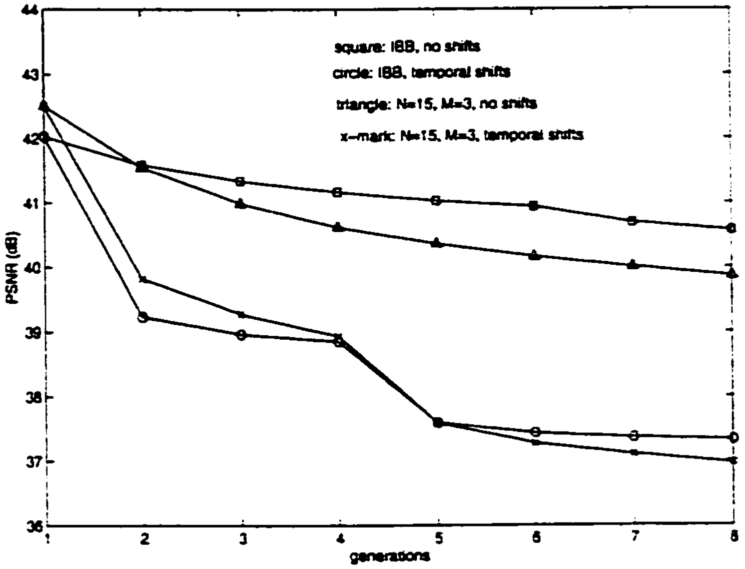


Figure 3.16: Effect of temporal shifts on multigeneration of the MPEG-2 4:2:2 profile. Table Tennis at 30Mb/s (Y-component). Temporal shifts occur after the first and fourth generations.

Flat and Non-flat Quantization Tables

Experiments show that flat quantization matrices perform, in general, better than non-flat quantization matrices in terms of multigeneration average PSNR. This is not surprising, as flat quantization matrices are known to statistically minimize the mean square error. However, the default non-flat quantization matrix given in the MPEG-2 standard is known to be subjectively optimized for the human visual system. As human subjective perception is less sensitive to errors in higher frequencies than those in lower frequencies, the default non-flat quantization matrix quantizes more coarsely the higher frequency DCT coefficients. This may result in better subjective quality of the picture even if flat quantization matrices are superior in terms of peak signal-to-noise ratio.

Table 3.5 compares between flat and non-flat quantization tables. Results are shown for Table Tennis with a GOP structure IBB and for bitrates 20, 30, and 40

Table 3.5: A comparison between flat and non-flat quantization tables. Multigeneration PSNR (Y-component) of Table Tennis using an IBB GOP structure.

Generation	20Mb/s		30Mb/s		40Mb/s	
number	flat	non-flat	flat	non-flat	flat	non-flat
1	38.91	38.19	42.03	40.98	44.57	43.87
2	38.72	38.01	41.88	40.74	44.03	43.27
3	38.68	37.99	41.81	40.70	43.66	42.95
4	38.65	37.97	41.70	40.67	43.44	42.74
5	38.62	37.96	41.61	40.66	43.25	42.61
6	38.60	37.95	41.47	40.65	43.09	42.52
7	38.58	37.95	41.39	40.65	42.97	42.45
8	38.56	37.95	41.34	40.64	42.88	42.40

Mb/s.

Type of Sequence

Multigeneration loss is sequence dependent and may differ significantly depending on the type of sequence. Results show that sequences which are harder to code due to their higher visual complexity suffer less degradation in multigeneration, or alternatively, sequences which start out at a relatively higher PSNR at the first generation, for a given bitrate, experience more degradation through multigeneration.

Table 3.6 shows the loss $L^{(8)}$ at the 8th generation for three types of sequences arranged from lower (left) to higher (right) complexity: Table Tennis, Flowers, and Mobile. Results are shown for a bitrate of 40Mb/s and for three GOP structures (I-only, IBB, and N=15, M=3). The loss is defined as

$$L^{(8)} = PSNR^{(1)} - PSNR^{(8)}$$

where $PSNR^{(1)}$ and $PSNR^{(8)}$ are the peak signal-to-noise ratios at the first and 8th generation, respectively. In the experiments, the same motion vectors of the first generation are used for each subsequent generations.

Table 3.6: Multigeneration loss (in dB) at 40Mb/s for different type of sequences (Y-component).

GOP structure	Multigeneration loss $L^{(8)}$		
	Tennis	Flowers	Mobile
I-only	1.59	0.35	0.30
IBB	1.69	1.23	0.47
N=15, M=3	3.41	2.57	0.62

Lower visual complexity video usually require less bits than high detailed sequences, for the same coding quality. Thus, for a given bitrate, blocks of low detailed video are quantized less coarsely than those of high detailed video. This is equivalent to having a smaller ratio between the DCT domain and the pixel domain quantizers for lower visual complexity video, which results in more PDQ (Pixel Domain Quantization) degradation [4, 3].

3.4 Summary of Chapter 3

This chapter presented five mechanisms that cause the continued degradation of picture quality in a multigeneration environment. Multigeneration error introduced by PDQ is larger for higher bit-rate video. It was observed that PDC mainly affects the DC of the DCT coefficients. For CCPV, it was noticed that using the quantization scaling parameters of the first generation results in an increase in the average peak signal-to-noise ratio (PSNR) of lower bit-rate video. For MVR, multigeneration error is reduced when the motion vectors of the first generation are used for each subsequent generation. For EPMC, multigeneration errors in reference pictures propagate to frames that are predicted from them. Consequently, EPMC multiplies the effect of other mechanisms. It also results in error propagation between predicted frames from generation to generation.

The experiments showed that PDQ and EPMC are the most important mechanisms for higher bitrates (lower compression ratios) while CCPV and MVR are more important for lower bitrates. The effect of PDC on multigeneration loss, for the sequences used in experiments, is very small. Other factors that affect multigeneration loss such as GOP structure, temporal shifts (changes in picture type), and visual complexity of sequences were also presented. It was observed that while long GOP structures have a higher PSNR at the first generation, they suffer however larger degradation throughout multigeneration than shorter GOP structures. Experiments showed that changes in pictures type during multigeneration introduce more errors and should be avoided if possible. Furthermore, it was also observerd that multigeneration loss is sequence dependent and that higher visual complexity sequences usually suffer less degradation in multigeneration.

Moreover, subjective evaluations were conducted by looking at the multigenerated video. High spatial frequency noise are introduced by multigeneration, especially in flat areas. The noise in B frames is higher than in P frames, while I frames are the least affected.

Chapter 4

Transcoding of MPEG Compressed Video

4.1 Introduction

The transmission of MPEG compressed video is an important issue in many of todays digital video services. While these services may use pre-encoded video, a lack of flexibility in transmission bit rates may arise due to changes in channel capacity or network demands. In situations such as network congestion, or when the transmission media has a lower capacity than the capacity required by the video bitstream, bit-rate reduction of the pre-encoded video is necessary. The process of converting a compressed video format into another compressed format is known as *transcoding*. This thesis addresses the specific transcoding problem of bit-rate reduction of a previously compressed MPEG video.

There are different approaches to the problem of bit rate conversion of MPEG compressed video. A straightforward approach is to fully decode the video, then to re-encode it at a lower bit rate. This approach has two disadvantages. First, the MPEG encoding algorithm usually requires high computational power. Thus, this approach is not an efficient solution in terms of implementation complexity, delay and cost. Second, errors are introduced due to the repeated compression/decompression of video [1, 4].

Another approach to bit rate conversion is to use scalable coding techniques [14, 21] where the video is coded as two or more layers, a base layer and one or more enhancement layers. The coded video information can be arranged such that the most important information is included in the base layer. In the case of network congestion, data from the lower priority layer maybe dropped to reduce the bitrate of the video. There are two disadvantages to this approach. First, scalability in MPEG-2 provides only a limited number of possible transmission bit rates. Second, in order to control transmission rates using layered coding, the encoder must

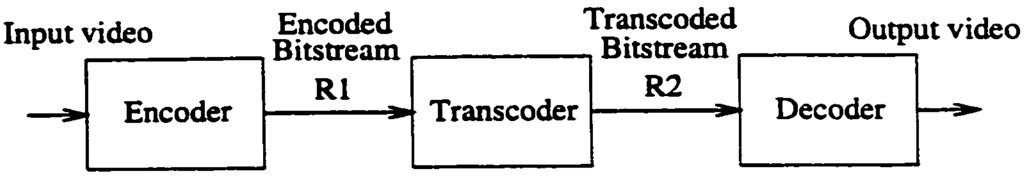


Figure 4.1: A basic video coding system including a transcoder.

take into account the congestion control policy of the network when a connection is set. This however may not be known when the video was first compressed for storage.

A more reliable approach is to transcode the compressed video to the desired bit rate. This is achieved by partial decoding of the bitstream, then performing the rate conversion by requantizing the DCT coefficients. This approach has two advantages over full decompression followed by re-compression of the video. First, since motion estimation is the most computationally intensive operation in the MPEG encoding algorithm, the complexity is significantly reduced by using the same motion vectors in transcoding. Second, this approach avoids some of the errors introduced when the video is fully decompressed and then compressed in a second generation [1, 4]. Figure 4.1 shows a basic block diagram of a video coding system that includes a transcoder. The encoder compresses an input video at a bit rate $R1$, then subject to certain constraints, the transcoder converts this compressed video at a bit rate $R2 < R1$. Next, the decoder decompresses the transcoded video bitstream for display.

The problem of transcoding has been studied by many researchers [73, 74, 75, 6, 76, 77, 5]. In [73], a two-stage coder for distribution of video at different bit rates is proposed. The first stage of the encoding is computationally intensive. All motion analysis are done in the first stage. The intermediate format resulting from the first stage is then stored and used as the source for distribution through diverse channels. While transcoding from the intermediate format to a desired bitstream is simpler and more efficient than the single-pass encoding, the intermediate format has a large overhead in storage capacity. In [74], a simple open loop transcoder architecture is presented. Rate-conversion of the compressed video is achieved by requantizing the DCT coefficients. The quantization step required for requantization is determined using the local and global quantization step of a picture. While the proposed requantization method is simple, it does not however take into account the propagation of errors (drift errors) which results from requantizing a motion-compensated compressed video. Another approach for high-quality transcoding is to use a closed loop technique that compensates for drift errors, as proposed in [75, 6, 76, 77]. Compensation for drift errors is performed either in the DCT domain [75, 6] or in the spatial domain [76, 77].

An important issue in bitrate conversion of compressed video is the optimization of the transcoder to improve picture quality. In [6], a Lagrangian rate-distortion optimization algorithm for bit reallocation in transcoding is presented. While the rate-distortion method improves the quality of the transcoded pictures, the optimization algorithm has added computational complexity and may impose extra delay in transcoding. The paper addresses this problem by using one-slice delay instead of one frame delay for a faster transcoder response to network demand. In [77], a detailed discussion of transcoding complexity and performance, including an analysis of the extra distortion introduced by requantization, is provided. It is shown that the complexity of the transcoder may be significantly less than a cascaded decoder/encoder system as several components in the cascaded decoder/encoder system may be combined such as motion vectors. The paper also provides experiments on the extra distortion introduced in transcoding due to the two cascaded quantizations: once in encoding, and once in transcoding. In [5], a study of the requantization problem in transcoding of MPEG-2 Intra frames with emphasis on designing the quantizer of the transcoder is presented. The paper provides two approaches for adjusting the decision levels of the transcoder's quantizer to improve picture quality. The first uses a mean-squared error (MSE) cost function, and the second uses a maximum a posteriori (MAP) cost function. Both MSE and MAP quantizers result in a higher picture quality in transcoding than the TM5 quantizer [39] In [78], a motion-vector refinement scheme to improve the quality of the transcoded pictures is presented. This paper also extends the motion-vector refinement scheme to cover the case when a frame-rate conversion is needed and proposes a motion vector composition method to compose an outgoing motion vector from the incoming motion vectors of the dropped frames.

4.2 Issues Related to Transcoding

In this thesis, two types of transcoders were used. The first type of transcoder uses a straightforward requantization method, without compensating for drift errors. The basic structure of a transcoder with no drift correction is shown in Figure 4.2. The advantages of this structure are its lower complexity and smaller memory requirements. This type of transcoder does not require any frame buffer. It is more suitable for low delay and low cost applications, and for video bitstreams having small group of pictures (GOP) structures, which reduce the effects of drift.

The second type of transcoder uses a feedback loop to compensate for drift errors. This method provides a higher quality transcoding, but with more complexity and memory requirements. The extra complexity added is due to the operations required for drift correction, while the increase in memory requirements is due to the frame buffers needed for drift compensation. Figure 4.3 shows the block diagram

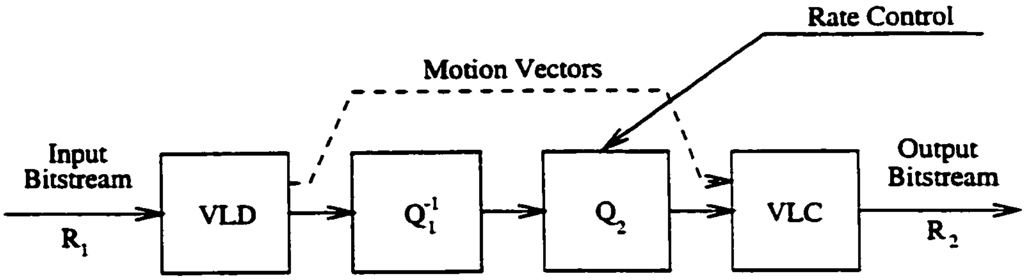


Figure 4.2: Transcoder structure with no drift correction.

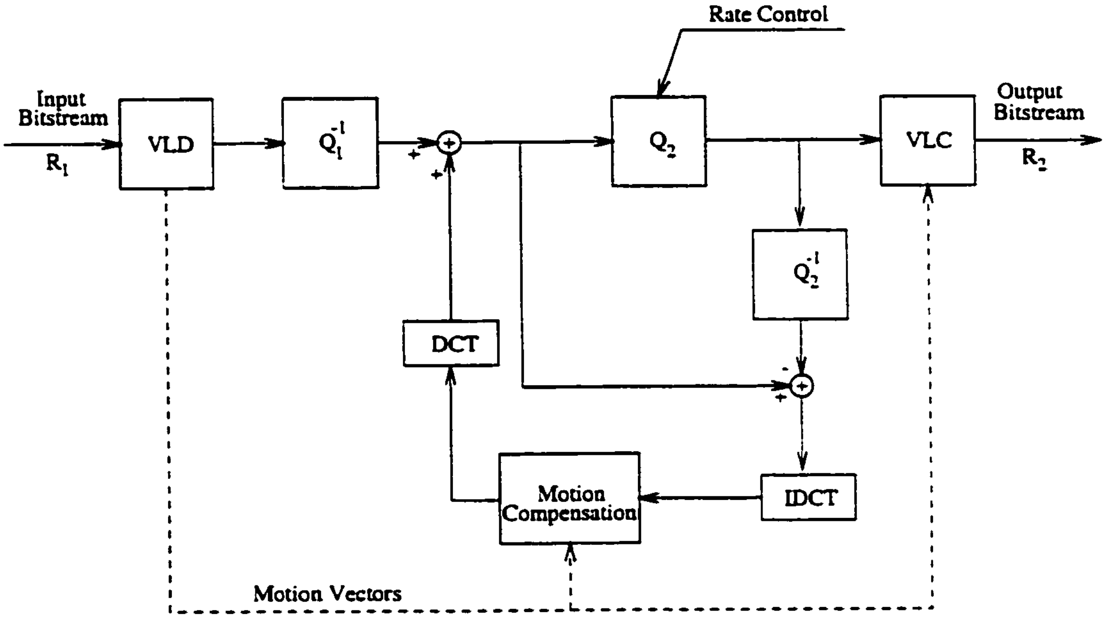


Figure 4.3: Transcoder structure with drift correction in the spatial domain.

of a transcoder with drift correction in the spatial domain. If drift correction is used for P and B picture coding types, then two frame buffers are needed. Memory requirements can be reduced to one frame buffer by performing drift correction only on P pictures, since requantization errors in B pictures do not propagate.

For both transcoding types, the requantization process is done on a macroblock basis. This reduces delay. Furthermore, the motion vectors of the input bitstreams are re-used. This avoids motion estimation, the most computationally intensive operation of the MPEG encoding algorithm. As expected, this significantly reduces the complexity of the system, as compared to a cascaded decoder-encoder system. Macroblock types may change after requantization. However, changes are limited. For example, a motion-compensated coded macroblock may be changed to a motion-compensated not coded macroblock if all the DCT coefficients are zero after requantization.

A brief discussion of requantization and rate control operations follows.

4.2.1 Requantization and Bit-rate Conversion

In MPEG standards, the DCT transform is performed on 8×8 blocks of data. The 64 DCT coefficients are uniformly quantized. the quantizer step size that is used for each DCT coefficient includes two components, a quantization coefficient which specifies a minimum step size for the particular DCT coefficient, and a quantization scaling parameter *qscale* used for bit-rate control. As *qscale* increases, the quantization of the DCT coefficients is coarser and consequently, the output bit rate decreases. In it's simplest case, The quantization process [75] can be expressed as

$$x_q(u,v) = Round\left(\frac{x(u,v)}{qscale \times w(u,v)}\right) \tag{4.1}$$

where $x(u,v)$ is the input DCT coefficient and $w(u,v)$ is its corresponding entry from the quantization table, $x_q(u,v)$ is the quantized DCT coefficient, and *Round* denotes integer division with rounding.

At the decoder, inverse quantization [75] may be performed as

$$y(u,v) = x_q(u,v) \times qscale \times w(u,v) \tag{4.2}$$

where $y(u,v)$ is the inverse quantized DCT coefficient.

Bit rate conversion can be performed in the compressed domain by requantization of the DCT coefficients. In its simplest case, this can be represented by

$$x_{q_2}(u,v) = Round\left(\frac{x_{q_1}(u,v) \times qscale_1 \times w_1(u,v)}{qscale_2 \times w_2(u,v)}\right) \tag{4.3}$$

where $x_{q_1}(u,v)$ is the input quantized DCT coefficient and $x_{q_2}(u,v)$ is its requantized version, $qscale_1$ is the input quantization scaling parameter and $qscale_2$ is the value required to meet the target bit rate. Usually, the quantization coefficients $w_1(u,v)$ (encoder) and $w_2(u,v)$ (transcoder) are the same and can be omitted from Equation 4.3. However, in some cases, the transcoder's quantization tables may be scaled in order to allow a suitable range for $qscale_2$ such that the target bit rate can be met. Note however that the above equation does not take into account the propagation of requantization errors (drift) due to motion compensation.

4.2.2 Rate Control

The rate control algorithm is not part of the standards and is thus left to the implementer to develop an efficient strategy. In [6], an approach to transcoding optimization and rate control by minimizing a Lagrangian cost has been proposed. This technique allows the selection of a trade-off point between rate and distortion subject to a rate constraint $R \leq R_{max}$. However, this method is computationally expensive and can not be easily implemented on a macroblock by macroblock basis. One well known rate control algorithm is described in the Test Model document, version 5 ($TM5$) [39]. The advantage of this algorithm is that the operations can be easily implemented on a macroblock basis, and therefore is more convenient for real time and low delay transcoding.

In this thesis, the bit rate control is based on the $TM5$. The algorithm consists of three steps, a *target bit allocation* which estimates the number of bits available to encode the next picture, a *rate control* that uses a "virtual buffer" to set the reference value of the quantization parameter for each macroblock, and an *adaptive quantization* which modulates the reference value of the quantization parameter according to the spatial activity in the macroblock and the average activity in the picture in order to derive the value of the quantization scaling parameter *qscale* used to quantize the macroblock.

Usually, the activity is computed in the pixel domain and is an important parameter in quantization, as it reflects the relative complexity of different macroblocks within the picture. Since the Human Visual System (HVS) is more sensitive to coding errors in the low-frequency regions of the picture (i.e., low-activity regions), these regions are usually quantized with a finer quantization step size. However, since local activity (over a macroblock) and average activity (over the picture) are not included in the encoded bitstream, they are estimated from local and average quantization parameters, respectively, as proposed in [74]. Specifically, the quantization scaling parameter for transcoding is expressed as:

$$qscale_2(j,k) = qscale_{ref}(j,k)\frac{qscale_1(j,k)}{\overline{qscale_1(k)}} \tag{4.4}$$

where $qscale_1(j,k)$ is the old quantization scaling parameter of the jth macroblock in the kth picture, $qscale_2(j,k)$ is its corresponding parameter for requantization, $qscale_{ref}(j,k)$ is the reference requantization parameter determined by the transcoder rate control algorithm, and $\overline{qscale_1(k)}$ is the average quantization parameter over the kth picture. The ratio $qscale_1(j,k)/\overline{qscale_1(k)}$ is the estimate of the normalized activity for the jth macroblock in the kth picture. In the experiments, the average quantization parameter of the last decoded picture of the same type is used for $\overline{qscale_1(k)}$.

4.2.3 Performance of Cascaded Quantization

Bit-rate conversion of compressed video involves two subsequent quantizations: the first in encoding, and the second in transcoding. In general, cascaded quantizations lead to an extra distortion as compared to direct quantization with the coarser quantizer. However, it is possible to use information available in the compressed video to reduce the requantization errors and achieve a higher transcoding quality.

In the following chapter, two methods for reducing requatization errors in transcoding are proposed [7, 8]. The methods are simple to implement and do not require side information.

4.3 Summary of Chapter 4

This chapter addressed the specific transcoding problem of bit-rate reduction of MPEG compressed video. The chapter discussed the importance of transcoding in providing flexibility in transmission bit-rates in situations such as network congestion, or when the transmission channel has a lower capacity than the capacity required by the pre-encoded video. Two types of transcoders with different complexity and memory requirements were discussed. The transcoder with no drift correction has a lower complexity and does not require frame buffers. It is more suitable for low delay applications and for video bitstreams having small GOP structures. The second type of transcoder has drift correction. It provides a higher quality, but with more complexity. In both transcoder structures, requantization is performed on a macroblock-by-macroblock basis. Moreover, the motion vectors pre-encoded in the compressed video are re-used. These reduces the delay and complexity of transcoding. The chapter also discussed issues related to transcoding such as requantization and bit-rate conversion of MPEG compressed video, and bit-rate control.

Chapter 5

Requantization for Transcoding of MPEG Compressed Video

5.1 Introduction

An important issue in bit-rate conversion is to provide requantization methods for efficient transcoding [5, 6]. In [5], two approaches for adjusting the decision levels of the transcoder's quantizer to improve transcoding quality are proposed. The first uses a mean-squared error (MSE) cost function, and the second uses a maximum a posteriori (MAP) cost function. Both methods uses a Laplacian pdf (probability density function) to model the distribution of the original DCT coefficients. The Laplacian parameter for each AC coefficient may be estimated from either the first generation coefficients or the original coefficients. In the latter case, the parameters have to be transmitted as additional side information. However, no algorithm was provided in [5] for estimating the Laplacian parameters from the first generation coefficients. In [6], a rate-distortion method is proposed using a Lagrangian optimization algorithm to derive the quantizer step size for transcoding each macroblock. This method improves picture quality. However, the optimization algorithm has added computational complexity and may impose extra delay in transcoding.

This chapter propose two methods to reduce the requantization errors in transcoding [7, 8]. The first method [7], presented in Section 5.2, assumes Laplacian distributions for the original AC coefficients of the DCT. A Laplacian parameter for each coefficient is estimated at the transcoder from the quantized input DCT coefficients. These parameters are then used in requantization to improve the quality of the transcoded video. The algorithm provided in this work to estimate the Laplacian parameters of the original DCT coefficients is simple to implement and may be adapted to other DCT-based coding schemes.

Section 5.3 presents selective requantization, a second method for improving the transcoding of MPEG compressed video [8]. The proposed method is based on avoiding critical ratios of the two cascaded quantizations (encoding versus transcoding) that either lead to larger transcoding errors or require a higher bit budget.

Both requantization methods are parts of the contributions of this thesis [7, 8].

5.2 Estimating Laplacian Parameters of DCT Coefficients in Transcoding of MPEG-2 Video

MPEG standards define how quantized DCT coefficients are reconstructed at the decoder, that is, the uniform reconstruction levels of the quantizer. A transcoder however may choose how to map the input DCT values onto a given set of reconstruction levels. Because the distribution of the AC coefficients of the DCT is not uniform, the uniform reconstruction of the input DCT coefficient at the transcoder, before requantization, may not be optimal. The requantization error may be reduced if the uniform reconstruction levels are replaced by the local centroid of the distribution of the DCT coefficients in the appropriate quantization interval. In general, while the DC coefficient has a uniform distribution [33] and is separately quantized using a uniform quantizer, the distribution of the AC coefficients may be represented by a two-sided Laplacian distribution [5, 33, 79]. The pdf $f(x)$ of the original AC coefficients x may be expressed as

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x|} \tag{5.1}$$

A pdf can be described for each AC coefficient once the corresponding Laplacian parameter α is known. This parameter may be estimated from the distribution of original DCT coefficients at the encoder, or from quantized DCT coefficients at the transcoder. The former however requires transmitting the set of parameters α with the compressed video and consequently introduces additional overhead. In this

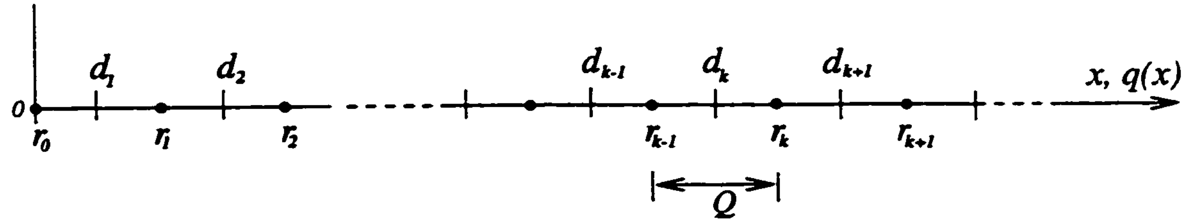


Figure 5.1: Decision and reconstruction levels of the midstep quantizer.

work, the Laplacian parameter α of the original distribution is estimated from the input DCT coefficients at the transcoder using maximum-likelihood estimation [80]. In the analysis, it is assumed that the encoder uses uniform quantization.

To develop the analysis for transcoding Intra macroblocks, consider a uniform midstep quantizer with no dead zone as shown in Figure 5.1. Quantization takes an input x and generates a level k. Assuming that the distribution of the DCT coefficients is symmetric around zero, only the magnitude value of the coefficient is considered. Thus, without loss of generality, k is positive. Let y be the reconstructed input DCT coefficients at the transcoder, i.e.,

$$y = kQ_1 \tag{5.2}$$

where Q_1 is the quantizer step size of the encoder. Note that Q_1 and k are transmitted in the encoded bitstream.

Let P_0 and P_k be the probability that $|x|$ lies in the intervals $[0, Q_1/2]$ for $k = 0$ and $(Q_1(k-1/2), Q_1(k+1/2)]$ for $k = 1, 2, \ldots$, respectively. Thus, P_0 and P_k are given by

$$\begin{aligned} P_0 &= 2\int_0^{Q_1/2} \frac{\alpha}{2} e^{-\alpha x} dx \\ &= 1 - e^{-\alpha Q_1/2} \end{aligned} \tag{5.3}$$

and

$$\begin{aligned} P_k &= 2\int_{Q_1(k-1/2)}^{Q_1(k+1/2)} \frac{\alpha}{2} e^{-\alpha x} dx \\ &= e^{-\alpha Q_1(k-1/2)} \left(1 - e^{-\alpha Q_1}\right) \\ &= e^{-\alpha(y-Q_1/2)} \left(1 - e^{-\alpha Q_1}\right) \qquad (5.4) \end{aligned}$$

Let $p(\mathbf{y}) = P(y_1, y_2, \ldots, y_N)$ be the joint probability distribution of N observed values of the sample y of the same frequency index over N input DCT blocks. In order to proceed with the analysis, independence assumption is needed. It is assumed that in a single frame of the same sequence the AC coefficients of the DCT, with the same frequency index, are independent samples of one realization. This is a reasonable assumption for high frequency DCT coefficients of different blocks in the picture. The independence assumption also increases the tractability of this model and is supported by the results presented in this chapter. Future work may also consider splitting frames into blocks and estimating the Laplacian parameters for each block, as well as addressing the Laplacian parameter estimation method in an object based coding context.

Assuming independent samples, $p(\mathbf{y})$ can be expressed as

$$p(\mathbf{y}) = \left(1 - e^{-\alpha Q_1/2}\right)^{N_0} \prod_{l=1, y_l \neq 0}^{N} e^{-\alpha(y_l - Q_1/2)} \left(1 - e^{-\alpha Q_1}\right) \qquad (5.5)$$

where N_0 is the number of zero DCT coefficients. A necessary condition for maximum-likelihood estimate [80] is

$$\frac{\partial}{\partial \alpha} \ln p(\mathbf{y}) \mid_{\alpha = \hat{\alpha}} = 0 \qquad (5.6)$$

This leads to

$$\frac{N_0 Q_1 e^{-\hat{\alpha} Q_1/2}}{2\left(1 - e^{-\hat{\alpha} Q_1/2}\right)} + \frac{M Q_1 e^{-\hat{\alpha} Q_1}}{\left(1 - e^{-\hat{\alpha} Q_1}\right)} - \sum_{l=1, y_l \neq 0}^{N} (y_l - Q_1/2) = 0 \qquad (5.7)$$

where M is the number of non-zero DCT coefficients. Let

$$A = \sum_{l=1, y_l \neq 0}^{N} (y_l - Q_1/2), \quad B = N_0 Q_1/2, \quad C = M Q_1$$

and

$$z = e^{-(\hat{\alpha}\frac{Q_1}{2})}$$

Equation 5.7 can be written as

$$\begin{aligned} A &= \frac{\left[N_0(Q_1/2)e^{-\hat{\alpha}Q_1/2}\right]\left[1 + e^{-\hat{\alpha}Q_1/2}\right] + MQ_1 e^{-\hat{\alpha}Q_1}}{(1 - e^{-\hat{\alpha}Q_1})} \\ &= \frac{Bz(1+z) + Cz^2}{(1-z^2)} \end{aligned} \tag{5.8}$$

The above yields

$$(A + B + C)z^2 + Bz - A = 0 \tag{5.9}$$

Note that the terms A, B, and C are computed from quantized DCT coefficients at the transcoder's input. Solving Equation 5.9, the maximum-likelihood estimate of the Laplacian parameter α for the midstep uniform quantizer is

$$\hat{\alpha} = -\frac{2}{Q_1} \ln z \tag{5.10}$$

Clearly, $\hat{\alpha}$ can be determined using only information available in the encoded bitstream. For illustration, the DCT coefficients of the original first frame of *Table Tennis* are quantized with a quantization step size $Q_1 = 16$ and the parameter $\hat{\alpha}$ is estimated for the the original DCT coefficient $X(0,2)$. The maximum-likelihood estimation yields $\hat{\alpha} = 0.06$. Figure 5.2 shows the actual histogram and the Laplacian distribution model (using $\hat{\alpha}$) of the DCT coefficient $X(0,2)$. Moreover, an empirical distribution function is obtained from the observations by sorting the data values in an increasing order and creating pairs $(X_i(0,2), i/N)$ where i is the index of the

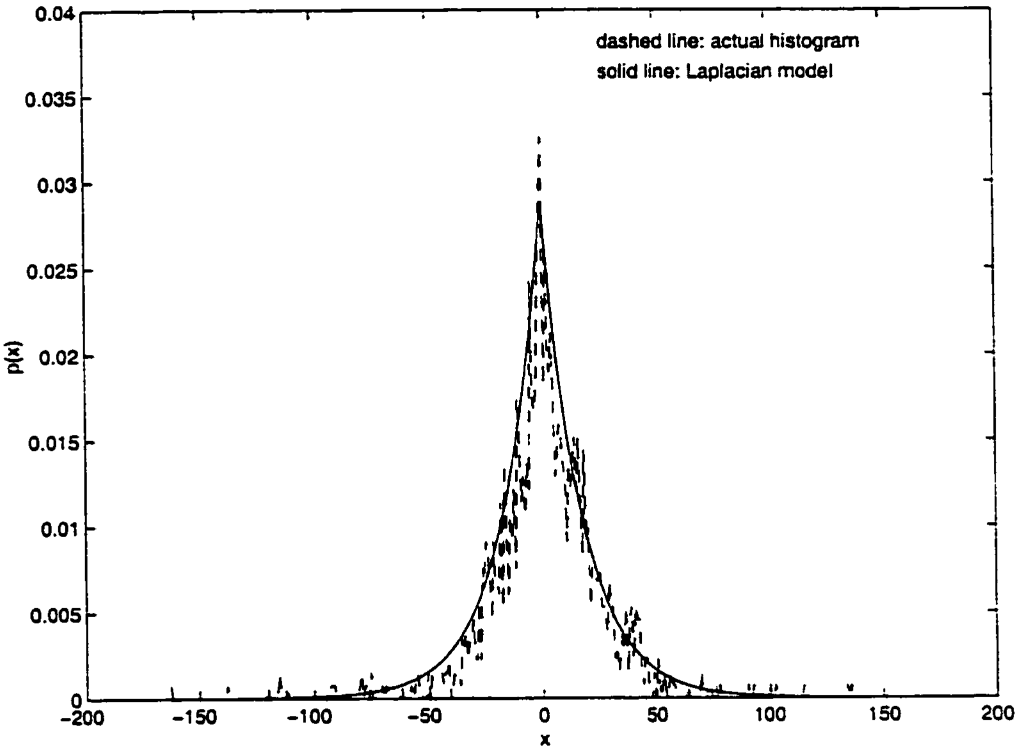


Figure 5.2: Actual histogram and Laplacian pdf model for the DCT coefficient $X(0,2)$ of the first frame (Y-component) of *Table Tennis* sequence, with an estimated parameter $\hat{\alpha} = 0.06$.

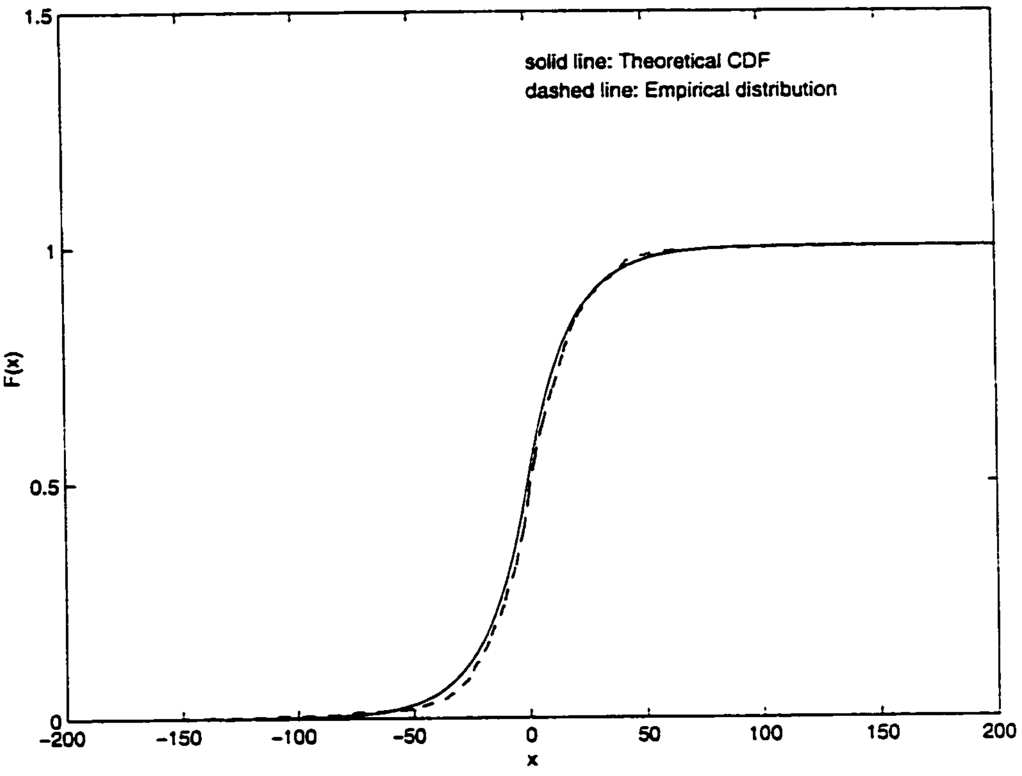


Figure 5.3: Empirical distribution and Laplacian CDF model of of the DCT coefficient $X(0, 2)$ of the first frame (Y-component) of *Table Tennis* sequence, with an estimated parameter $\hat{\alpha} = 0.06$.

data value $X_i(0,2)$ in the sorted list. The empirical distribution is plotted versus the theoretical Cumulative Distribution Function (CDF) and is shown in Figure 5.3.

To develop similar analysis for transcoding non-Intra macroblocks, consider a uniform midriser quantizer with no dead-zone. Note however that the analysis can be easily extended to include quantization with a dead zone around zero. The reconstructed input DCT coefficients at the transcoder can be expressed as

$$y = (k + \frac{1}{2})Q_1 \tag{5.11}$$

Let P_k be the probability that the magnitude of the DCT coefficient lies in the intervals $(kQ_1, (k+1)Q_1]$ for $k = 0, 1, 2, \ldots$, therefore;

$$\begin{aligned} P_k &= 2\int_{kQ_1}^{(k+1)Q_1} \frac{\alpha}{2} e^{-\alpha x} dx \\ &= e^{-\alpha(y-Q_1/2)}\left(1 - e^{-\alpha Q_1}\right) \end{aligned} \tag{5.12}$$

Analogous to Equation 5.5, the joint probability distribution $p(\mathbf{y})$ of N observed sample values of y for the midriser quantizer is

$$p(\mathbf{y}) = \prod_{n=1}^{N} \left[e^{-\alpha(y_n - Q_1/2)}\left(1 - e^{-\alpha Q_1}\right)\right] \tag{5.13}$$

Applying the condition for maximum-likelihood estimation given by Equation 5.6 yields:

$$\frac{NQ_1 e^{-\hat{\alpha} Q_1}}{(1 - e^{-\hat{\alpha} Q_1})} - \sum_{n=1}^{N}(y_n - Q_1/2) = 0 \tag{5.14}$$

Let

$$A = \sum_{n=1}^{N}\left(y_n - \frac{Q_1}{2}\right), \quad B = NQ_1, \quad \text{and} \quad z = e^{-\hat{\alpha} Q_1}$$

Equation 5.14 can be written as

$$(A + B)z - A = 0 \tag{5.15}$$

Solving for z, the maximum-likelihood estimate of the Laplacian parameter for the midriser quantizer is computed from

$$\hat{\alpha} = -\frac{1}{Q_1} \ln z \tag{5.16}$$

Note that the parameter $\hat{\alpha}$ has to be estimated for each AC coefficient and for each of the luminance (Y) and chrominance (Cb and Cr) components as they have different distributions. Once $\hat{\alpha}$ is known, and prior to requantization with a coarser step size Q_2, the transcoder reconstructs the input DCT coefficients at the local centroid of the distribution instead of the uniform reconstruction of Equation 5.20.

If d_k is the kth decision level of the encoder quantizer, then for a given AC coefficients x in the interval $(d_k, d_{k+1}]$, the local centroid [26] is defined as

$$\begin{aligned} c_k &= \frac{\int_{d_k}^{d_{k+1}} x f(x) dx}{\int_{d_k}^{d_{k+1}} f(x) dx} \\ &= \frac{\int_{d_k}^{d_{k+1}} x \frac{\alpha}{2} e^{-\alpha x} dx}{\int_{d_k}^{d_{k+1}} \frac{\alpha}{2} e^{-\alpha x} dx} \\ &= \frac{\left(d_k + \frac{1}{\alpha}\right) e^{-\alpha d_k} - \left(d_{k+1} + \frac{1}{\alpha}\right) e^{-\alpha d_{k+1}}}{\left(e^{-\alpha d_k} - e^{-\alpha d_{k+1}}\right)} \end{aligned} \tag{5.17}$$

Note that the above equation is one of the Lloyd-Max quantizer design rules [26]. The stepsize of the encoder's uniform quantizer can be expressed in terms of its decision levels as

$$Q_1 = d_{k+1} - d_k \tag{5.18}$$

From Equations 5.17 and 5.18, the local centroid of the DCT coefficients is computed at the transcoder as

$$\begin{aligned} c_k &= \frac{\left(d_k + \frac{1}{\hat{\alpha}}\right) - \left(d_k + Q_1 + \frac{1}{\hat{\alpha}}\right) e^{-\hat{\alpha} Q_1}}{\left(1 - e^{-\hat{\alpha} Q_1}\right)} \\ &= d_k + \frac{1}{\hat{\alpha}} - \frac{Q_1 e^{-\hat{\alpha} Q_1}}{1 - e^{-\hat{\alpha} Q_1}} \end{aligned} \tag{5.19}$$

Clearly, c_k depends on the step size and decision level d_k of the encoder's quantizer, and on the estimated Laplacian parameter $\hat{\alpha}$.

5.2.1 Experiments

Transcoding is achieved by requantizing the input DCT coefficients. The following experiments use transcoding with drift correction. The transcoder structure is shown in Figure 4.3. For constant bit-rate (CBR) transcoding, the $TM5$ rate control algorithm [39] is used, while for variable bit-rate (VBR) transcoding, the rate control is disconnected. Since estimating the Laplacian parameters $\hat{\alpha}$ for a current frame imposes a delay of one frame, this requirement is removed by using the estimated parameters of the most recently transcoded frame of the same type, since usually image statistics do not vary much over a short period of time. In CBR transcoding, the average step size over the frame is used instead of Q_1 for the terms B, C, and z (Equations 5.9 and 5.15). Moreover, in quantization of non-intra macroblocks, a uniform quantizer with a dead-zone around zero is used similar to MPEG-2 standard. For VBR and CBR Intra-frame transcoding, two experiments were performed using 150 frames (each 352×240) of *Flower Garden* sequence. Figure 5.4 shows the PSNR (Y-component) and bits versus frame number for VBR transcoding of *Flower Garden* for quantization step sizes $Q_1 = 16$ and $Q_2 = 30$. In normal transcoding, input DCT coefficients are uniformly reconstructed before requantization, following the MPEG-2 uniform reconstruction equation. Notice the improvement in PSNR and bits achieved by using the Laplacian model in transcoding. The average increase in PSNR in this example is respectively $0.3, 0.9$, and 1.7 dB per frame for the Y, Cb, and Cr components. Furthermore, the number of bits spent by using this model is reduced by 26% as compared to normal transcoding. The equivalent bit rate for

Table 5.1: Intra-frame transcoding of *Flower Garden* and *Football*. PSNR for transcoding using a Laplacian pdf model (PSNR$_L$) and for normal transcoding (PSNR$_N$). The input bit rate is 8Mb/s.

Bit Rate	PSNR$_L$ (dB)	PSNR$_N$ (dB)	Δ (dB)
Flower Garden			
7	29.7	27.3	2.4
6	28.0	26.7	1.3
5	26.9	26.6	0.3
4	26.5	26.2	0.3
Football			
7	35.8	34.1	1.7
6	34.4	33.6	0.8
5	33.9	33.5	0.4
4	33.5	33.4	0.1

the input bitstream is 8.5Mb/s. Likewise, the equivalent output bit rate for normal transcoding and transcoding using Laplacian pdf model is 7.3Mb/s and 5.4Mb/s, respectively.

Figure 5.5 shows the PSNR (Y-component) versus frame number for CBR transcoding of *Flower Garden* for input and output bit rates of 8Mb/s and 7Mb/s, respectively. The average increase in PSNR by using the Laplacian model is respectively 2.4, 2.5, and 2.8 dB per frame for the Y, Cb, and Cr components. Table 5.1 shows average PSNR of Intra-frame transcoding of *Flower Garden* and *Football* for an input bit rate of 8Mb/s and for different output bit rates. Two cases are shown: transcoding using a Laplacian pdf model (PSNR$_L$) and normal transcoding (PSNR$_N$). The parameter Δ denotes the difference between PSNR$_L$ and PSNR$_N$. Note that the improvement achieved using the Laplacian model differs from sequence to sequence and for different bit rates.

For both Intra-frame and Inter-frame CBR transcoding, Table 5.2 shows the PSNR of normal transcoding and of transcoding using a Laplacian pdf model. The sequences *Flower Garden* (Flr.), *Football* (Ftb.), *Table Tennis* (Ten.), *Miss America*

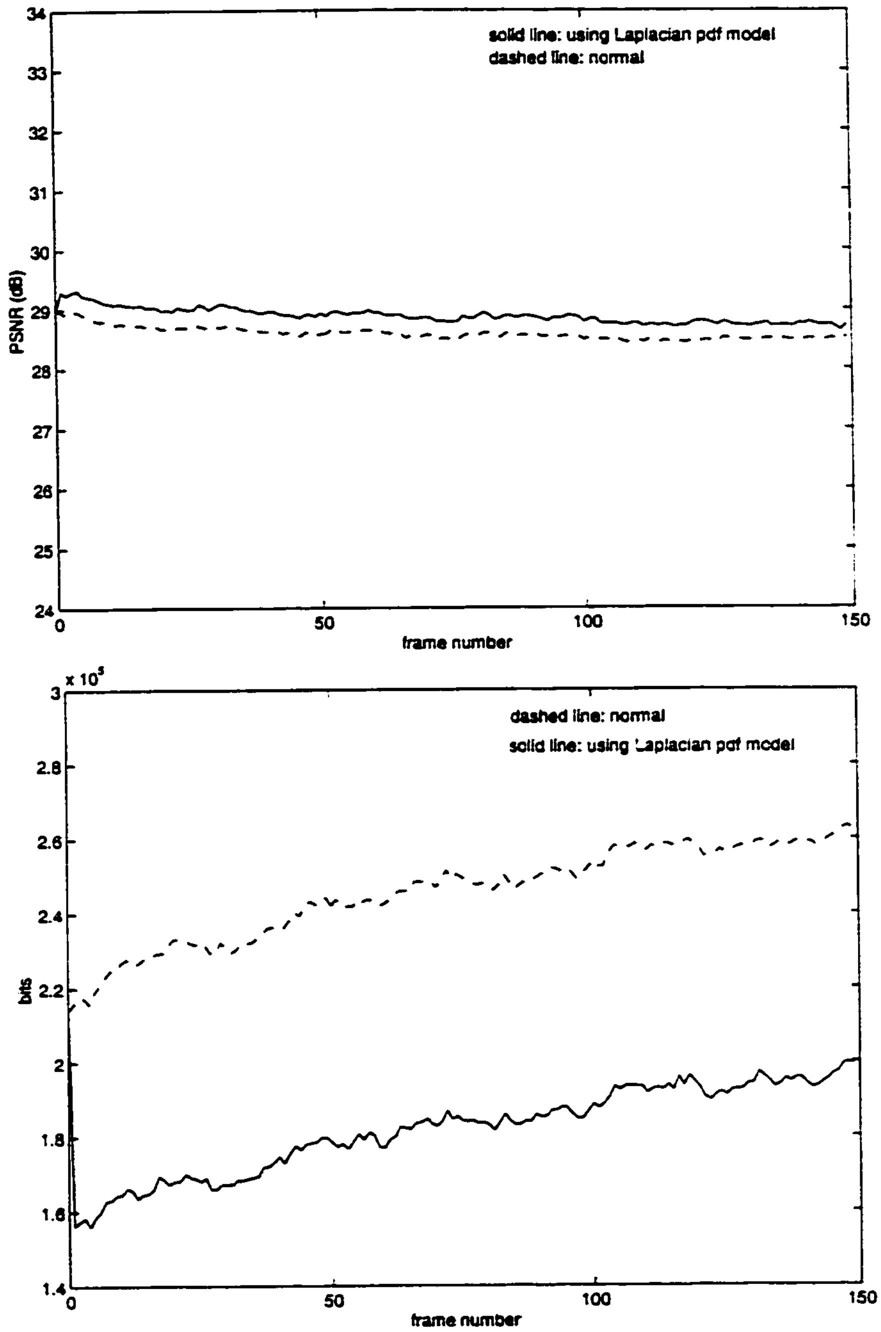


Figure 5.4: VBR Intra-frame transcoding using a Laplacian pdf for the DCT coefficients. PSNR (top) and bits (bottom) versus frame number of *Flower Garden* for $Q_1 = 16$ and $Q_2 = 30$.

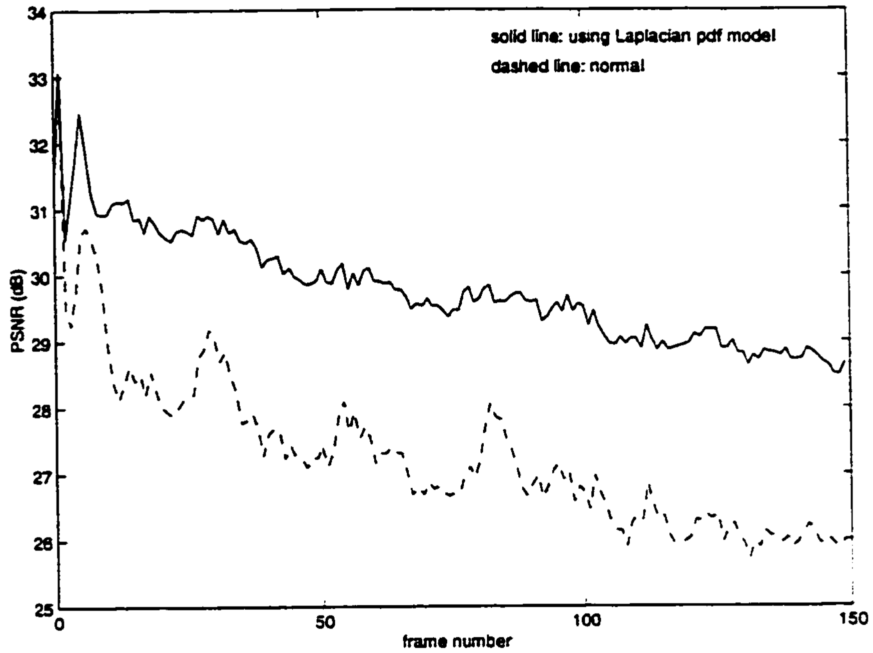


Figure 5.5: CBR Intra-frame transcoding using a Laplacian pdf for the DCT coefficients. PSNR versus frame number for *Flower Garden* for input and output bit rates of 8Mb/s and 7Mb/s, respectively.

(Mis.), and *Salesman* (Slm.) are used, with a group of pictures (GOP) structure N=15, M=3. The input and output bit rates for transcoding are respectively 2Mb/s and 1.5Mb/s. Notice that transcoding using the Laplacian pdf model outperforms normal transcoding.

Next section presents selective requantization [8], a simple method for reducing requantization errors in transcoding.

Table 5.2: PSNR for transcoding using a Laplacian pdf model (PSNR_L) and for normal transcoding (PSNR_N) of five video bitstreams with a GOP structure N=15, M=3.

Transcoding PSNR (dB)					
Case	Flr.	Ftb.	Ten.	Slm.	Mis.
PSNR_L	26.1	30.0	33.8	38.3	41.3
PSNR_N	25.8	29.6	33.4	38.0	40.9
Δ	0.3	0.4	0.4	0.3	0.4

5.3 Selective Requantization

As stated earlier, bit-rate conversion of compressed video involves two subsequent quantizations: the first in encoding, and the second in transcoding. In general, cascaded quantizations lead to an extra distortion as compared to direct quantization with the coarser quantizer. This extra distortion is denoted as "cascading error". The cascading error depends on the *ratio* between the finer quantizer (encoder) and the coarser quantizer (transcoder). Thus, as the step size of the finer quantizer is encoded in the bitstream and is available to the transcoder, it is possible to use this information to reduce the requantization errors and achieve a higher transcoding quality.

For simplicity, consider the quantization of a non-negative input data value x using a uniform midstep quantizer (suitable for Intra macroblocks) with no dead zone, as shown in Figure 5.6. Input-output characteristics of the quantizer are mirrored for negative x. The quantizer is completely specified by a set of reconstruction levels, r_k, and a set of decision levels, d_k, given by

$$r_k = kQ, \quad \text{and} \quad d_k = \frac{r_k + r_{k-1}}{2} = (k - \frac{1}{2})Q \tag{5.20}$$

where Q is the step size of the quantizer and k is an integer representing the level. Thus, given a value x and its reconstruction $q(x) = kQ$, The quantization error, ε, can be expressed as

$$\varepsilon = x - q(x) \tag{5.21}$$

Let Q_1 and Q_2, where $Q_1 < Q_2$, be the step sizes of the finer quantizer (encoder) and coarser quantizer (transcoder), respectively. To illustrate the idea of selective quantization, an experiment is performed using 150 frames (each 352 × 240) of *Flower Garden*. and only Intra frames coding. Figures 5.7.(a)., 5.7.(b)., and

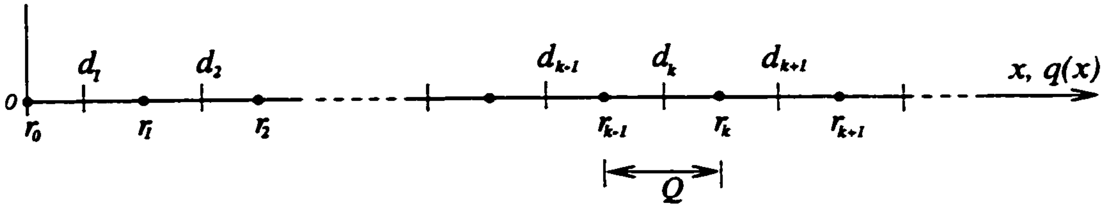


Figure 5.6: Decision and reconstruction levels of the midstep quantizer.

5.7.(c). show the rate-distortion (*R-D*) curves for transcoding of *Flower Garden*, originally encoded at $Q_1 = 4$, $Q_1 = 8$ and $Q_1 = 24$, respectively. Clearly, from the set of operating *R-D* points, some of these points (and their corresponding Q_2) will not fall on the convex hull of the overall *R-D* characteristic and therefore they are not optimal in the rate-distortion sense.

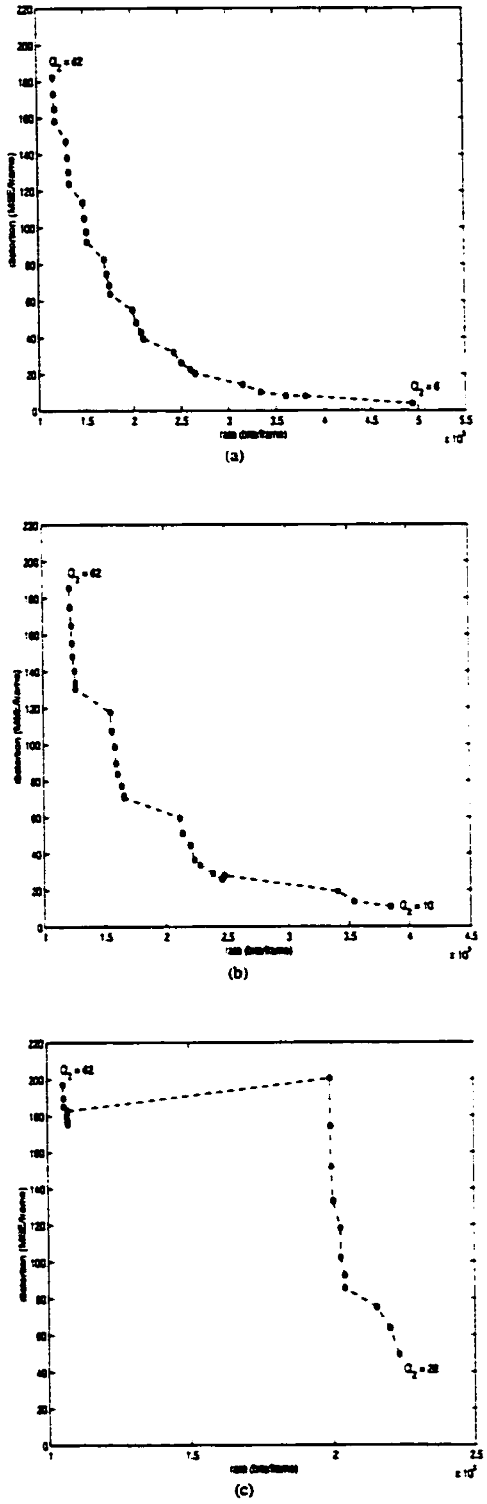
Let $\varepsilon_{1,2}$ be the total quantization error that results from quantizing with step size Q_1 followed by step size Q_2, and let ε_2 be the error resulting from direct quantization with step size Q_2. The cascading error ε_c is defined as

$$\varepsilon_c = \varepsilon_{1,2} - \varepsilon_2 \tag{5.22}$$

In general, the cascading error increases, on average, when many reconstruction levels from the finer quantizer are aligned with decision levels from the coarser quantizer. The maximum cascading error ε_{cmax} depends on the ratio Q_2/Q_1. The upper bound of ε_{cmax} is equal to the step size of the finer quantizer Q_1. This corresponds to the special case illustrated in Figure 5.8 where the original data value lies on a decision boundary of the finer quantizer, and the respective reconstruction level of the finer quantizer lineup with a decision level of the coarser quantizer. Thus, the magnitude of maximum cascading error is bounded by

$$0 \leq |\varepsilon_{cmax}| \leq Q_1 \tag{5.23}$$

If the finer quantizer's cell, corresponding to the original data value, is totally contained in a cell of the coarser quantizer, then cascaded quantization gives the same



(a)

(b)

(c)

Figure 5.7: Rate-distortion curves of *Flower Garden* (Intra-frame coding). The sequence is originally encoded at the following values of Q_1: (a) $Q_1 = 4$. (b) $Q_1 = 8$. (c) $Q_1 = 24$.

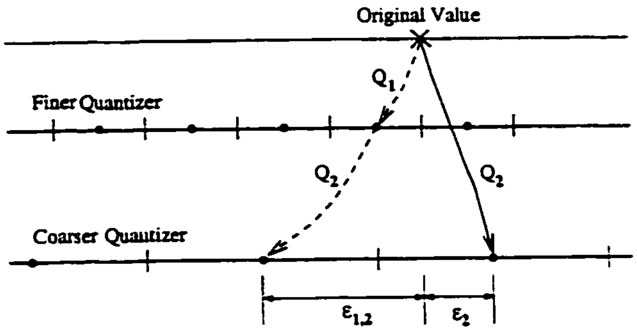


Figure 5.8: A situation where maximum cascading error equals Q_1.

result as direct quantization with Q_2, i.e., no cascading error is introduced by requantization (Figure 5.9.(a)). However, if the finer quantizer's cell overlap with two cells of the coarser quantizer, then cascading error is introduced by requantization if the reconstructed data value after the first quantization and the original data value each fall into a different quantization cell in the coarser quantizer (Figure 5.9.(b)).

To study how the ratio Q_2/Q_1 affects requantization, for a uniform midstep quantizer, let $r_{k_1} = k_1 Q_1$ be the reconstruction levels of the finer quantizer, and let $d_{k_2} = (k_2 - 1/2)Q_2$ be the decision levels of the coarser quantizer. The condition that a reconstruction level from the finer quantizer is aligned with a decision level from the coarser quantizer is given by

$$k_1 = \left(\frac{2k_2 - 1}{2}\right)\frac{Q_2}{Q_1} \tag{5.24}$$

Equation 5.24 is satisfied if there exists a ratio Q_2/Q_1 and an integer k_2 that result in an integer k_1. Clearly, an even integer ratio of Q_2/Q_1 satisfies this condition. This is illustrated in Figure 5.11.(a). The magnitude of maximum cascading error usually reaches its highest bound (Equation 5.23) when the ratio Q_2/Q_1 is an even integer. Notice that Equation 5.24 may be also satisfied for non-integer ratios of Q_2/Q_1. An example of that is the ratio $Q_2/Q_1 = 8/6$. Furthermore, the percentage of reconstruction levels from the finer quantizer that are aligned with decision levels

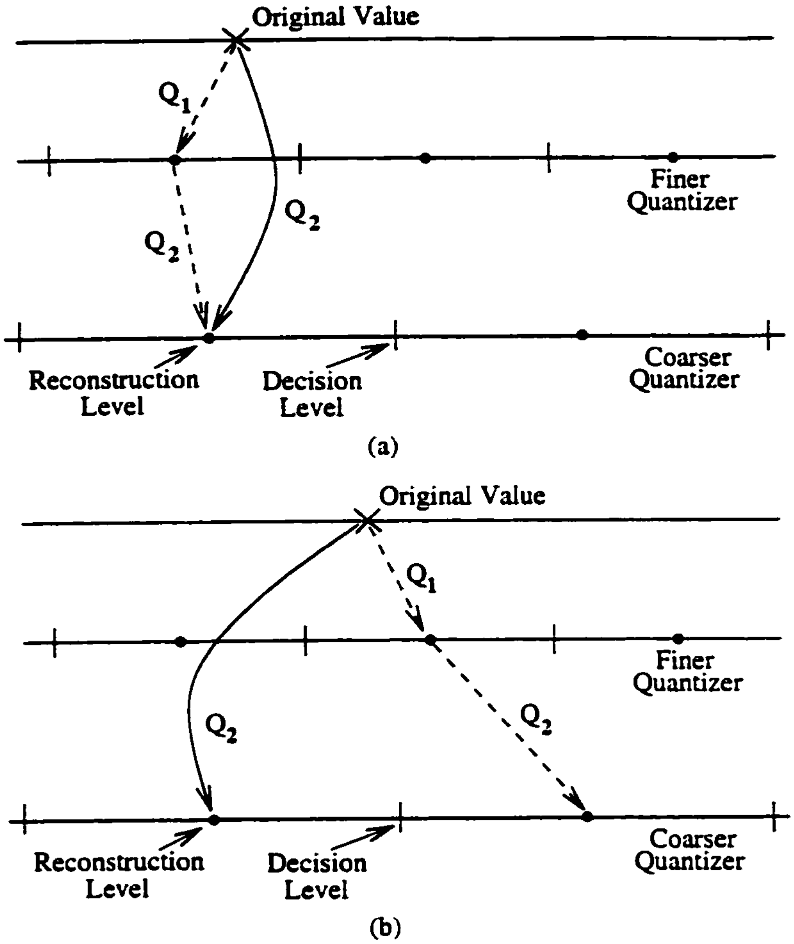


Figure 5.9: Cascaded quantizations (dashed arrows) and direct quantization (solid arrow). (a) Cascaded quantizations yield the same results as direct quantization. (b) Cascaded quantizations introduce extra distortion, as compared to direct quantization.

from the coarser quantizer depends on the ratio Q_2/Q_1. This percentage may be computed off-line. For example, if $Q_2/Q_1 = 2$, then half of the reconstruction levels from the finer quantizer are aligned with decision levels from the coarser quantizer. This can be seen in Figure 5.10 for $Q_1 = 6$ and the case when $Q_2 = 12$. The experiments show that a ratio of 2 for Q_2/Q_1 usually yields larger transcoding errors. Moreover, large drops in bits usually occur at values of Q_2 that are integer multiple of $2Q_1$. This is because more requantized DCT coefficients are mapped to

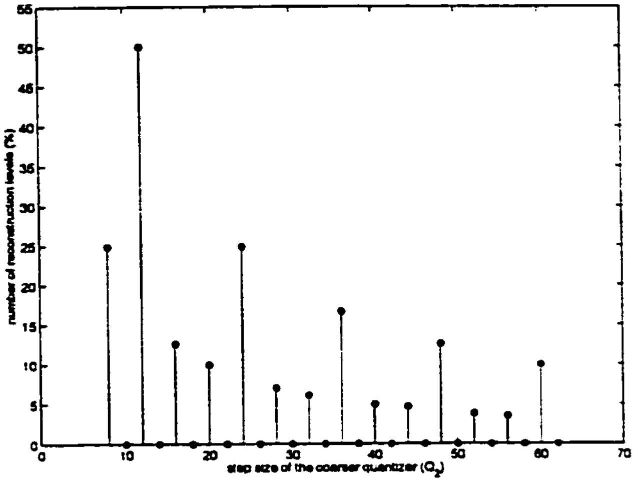


Figure 5.10: Number of reconstruction levels (in percentage) from the finer quantizer that lineup with decision levels from the coarser quantizer, for $Q_1 = 6$ and Q_2 ranging from 8 to 62 with intervals of 2.

lower quantization levels, and consequently, they are coded with fewer bits. This is illustrated in Section 5.3.1.

Equation 5.24 is not satisfied for all integers k_2 and an odd integer ratio for Q_2/Q_1. In this case, the cells of the finer quantizer are totally contained inside the coarser quantizer cells and consequently, the cascading error is zero. This is illustrated in Figure 5.11.(b). Thus, given a ratio Q_1/Q_2, the value of $|\varepsilon_{cmax}|$ for that ratio can be determined from the position of decision levels of the finer quantizer relative to the coarser quantizer. For example, Figure 5.12 shows the absolute value of the maximum cascading error that may occur for $Q_1 = 6$ and Q_2 ranging from 8 to 62 with intervals 2. Notice that the values of $|\varepsilon_{cmax}|$ satisfy Equation 5.23.

Using similar analysis for a midriser quantizer (suitable for non-Intra Macroblocks), let $r_{k_1} = (k_1 + 1/2)\, Q_1$ be the reconstruction levels of the finer quantizer, and let $d_{k_2} = k_2 Q_2$ be the decision levels of the coarser quantizer. The equation

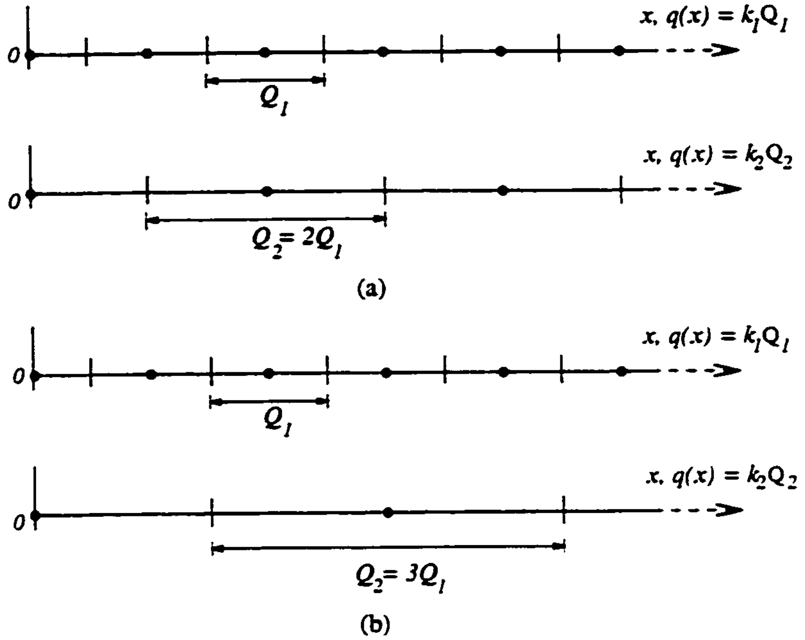


Figure 5.11: Ratio of coarser quantizer (Q_2) to finer quantizer (Q_1), for midstep quantizers. (a) Even integer ratio ($Q_2/Q_1 = 2$). (b) Odd integer ratio ($Q_2/Q_1 = 3$).

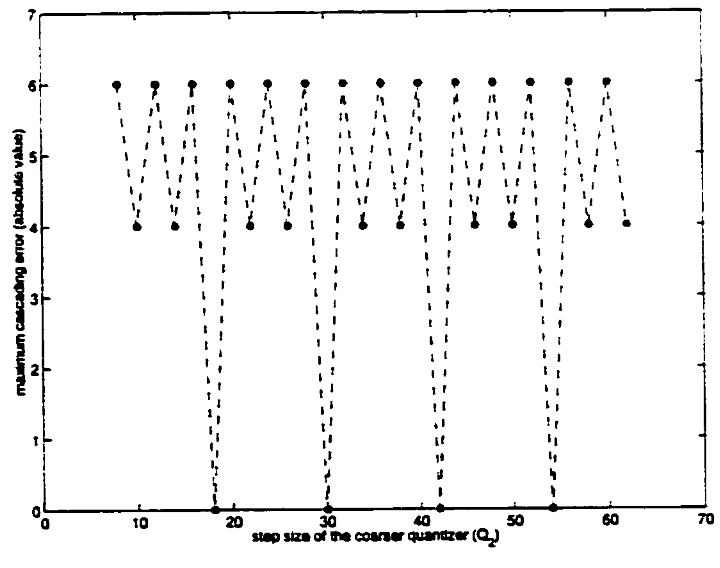


Figure 5.12: Absolute value of maximum cascading error (ε_{cmax}) for $Q_1 = 6$ and Q_2 ranging from 8 to 62 with intervals of 2.

analogous to Equation 5.24 is given by

$$(2k_1 + 1) = 2k_2\frac{Q_2}{Q_1} \tag{5.25}$$

Clearly, Equation 5.25 is satisfied for any odd integer ratio of $2Q_2/Q_1$ and odd integers k_2. In this case, additional loss is usually introduced by cascading quantizations. For example, if $Q_1 = 4$ and $Q_2 = 6$, then one-third of the reconstruction levels of the finer quantizer are aligned with decision levels from the coarser quantizer. The experiments show that a ratio of 3 for 2 Q_2/Q_1 may lead to larger transcoding errors. On the other hand, Equation 5.25 is not satisfied for even integer ratios of $2Q_2/Q_1$ and all integers k_2. In this case, direct quantization with Q_2 and cascaded quantizations with Q_1 followed by Q_2 yield the same quantization error.

In this thesis, the selective requantization method is applied to CBR transcoding of MPEG-2 compressed video. A linear mapping for *qscale* (i.e., *q_scale_type* = 0) was used in the experiments, however, the algorithm maybe extended to cover non-linear mapping for *qscale*. For simplicity, it is assumed that the transcoder is using the same quantization matrices that are encoded in the bitstream. The algorithm for selective requantization includes in its code the following conditions

```
1. Intra macroblock
   (a) $if((qscale_2\%(2 * qscale_1)) == 0)$
          $qscale_2 = qscale_2 + 2;$
   (b) $if(((qscale_2 + 2)\%qscale_1) == 0)$
          $if((((qscale_2 + 2)/qscale_1)\%2)! = 0)$
            $qscale_2 = qscale_2 + 2;$
   (c) $if(((qscale_2 + 2)\%(2 * qscale_1)) == 0)$
          $qscale_2 = qscale_2 + 4;$
```

```
2. Non-Intra macroblock
    (a) $if(((qscale_2 + 2)\%qscale_1) == 0)$
            $qscale_2 = qscale_2 + 2;$
    (b) $if(((2 * qscale_2)\%qscale_1) == 0)$
            $if((((2 * qscale_2)/qscale_1)\%2)! = 0)$
              $qscale_2 = qscale_2 + 2;$
```

where the above mathematical operators are similar to those used in the C programming language. In summary, the above items are the result of the previous analysis for midstep (Intra) and midriser (Non-Intra) quantizers. For Intra macroblocks, Item 1.(a) avoids even integer ratios of Q_2/Q_1. These ratios usually lead to larger transcoding errors. Item 1.(b) selects the next higher $qscale_2$ if it results in an odd integer ratio of Q_2/Q_1. The new value for Q_2 yields zero cascading error. Item 1.(c) reduces the number of bits needed to code Intra macroblocks in an MPEG-like coding scheme without a significant increase in errors (see Section 5.3.1). For Non-Intra macroblocks, Item 2.(a) selects the next higher $qscale_2$ if it results in an even integer ratio of $2Q_2/Q_1$. This ratio has zero cascading error. Item 2.(b) avoids odd integer ratios of $2Q_2/Q_1$. These ratios usually leads to larger transcoding errors. The experiments show that Items 1.(c) and 2.(a) are the most important for Intra and non-Intra macroblocks, respectively, and for the sequences used in this thesis. A practical example that illustrates the above conditions is shown in Table 5.3. The initial values of Q_2 set by the rate control are shown against the values of Q_2 in selective quantization for both midstep and midriser quantizers and for $Q_1 = 8$.

Table 5.3: Practical example to illustrate Selective Requantization.

Midstep Quantizer, $Q_1 = 8$															
Q_2 (initial)	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
Q_2 (Selective)	8	10	12	18	18	18	20	24	24	26	28	34	34	34	36
Midriser Quantizer, $Q_1 = 8$															
Q_2 (initial)	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
Q_2 (Selective)	8	10	14	16	16	18	22	24	24	26	30	32	32	34	38

5.3.1 Experiments

The following experiments use transcoding with drift correction. The transcoder structure is shown in Figure 4.3. For constant bit-rate (CBR) transcoding, the $TM5$ rate control algorithm [39] is used, while for variable bit-rate (VBR) transcoding, the rate control is disconnected.

For VBR Intra-frames transcoding, 150 frames (each 352 × 240) of *Flower Garden* were used. The sequence is first encoded with an MPEG-2 software encoder and a constant step size $Q_1 = 16$. The resulting bitstream is then transcoded at different values of Q_2 ranging from 18 to 40, with intervals of 2. In direct encoding, the sequence is encoded directly with Q_2. The peak signal-to-noise ratio (PSNR) is used as an objective measure of the video quality, with the original sequence as reference. Figure 5.13 shows transcoding average PSNR and bit budget versus Q_2. Notice the PSNR and bit budget for $Q_2 = 32$ and $Q_2 = 34$. The former results in a ratio $Q_2/Q_1 = 2$, which usually leads to larger requantization errors. The PSNR for $Q_2 = 34$ is higher by 0.3 dB with 0.01% bits lower than in the case of $Q_2 = 32$. Furthermore, notice the PSNR and bit budget for $Q_2 = 30$ and $Q_2 = 34$. Both values of Q_2 yield almost the same PSNR. However, the reduction in bits by choosing $Q_2 = 34$ over $Q_2 = 30$ is about 39%. This is important in CBR transcoding where the saving in bits can be used on coding other macroblocks such that the overall video quality is improved.

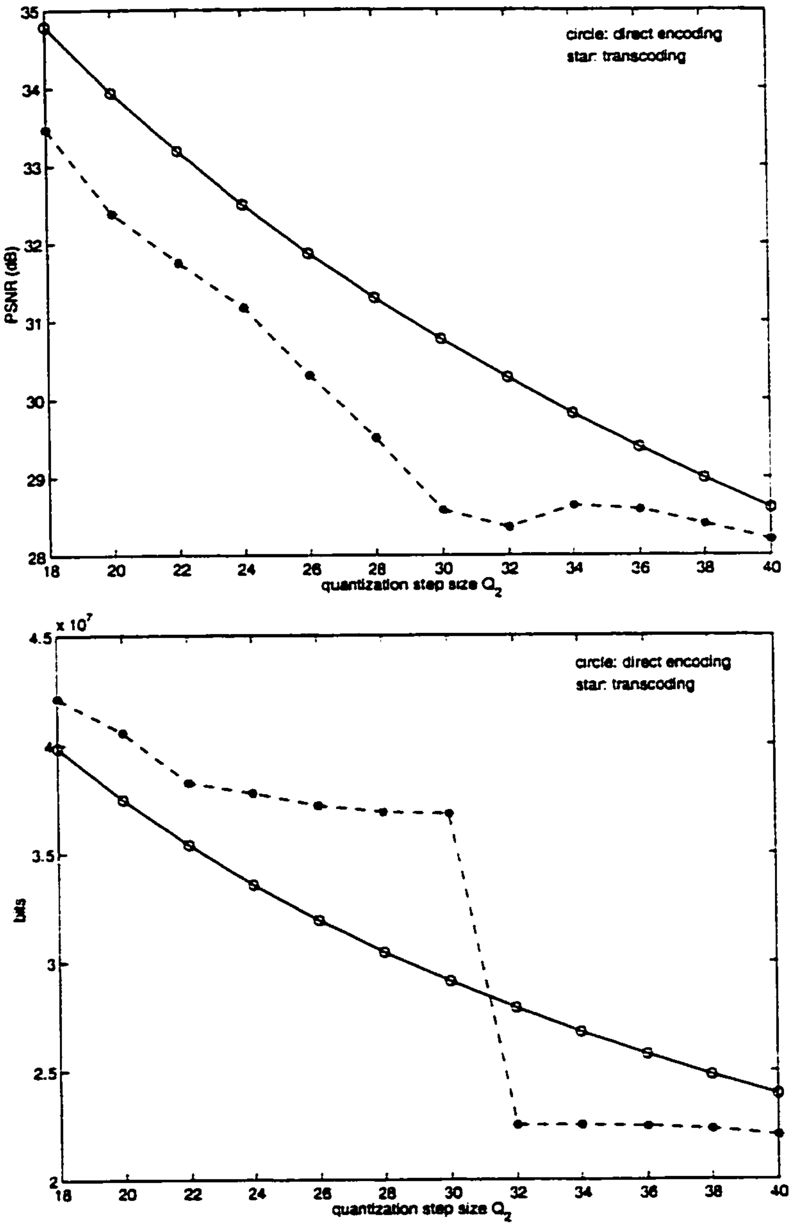


Figure 5.13: Intra-frame Transcoding average PSNR (top) and bit budget (bottom) of *Flower Garden* for $Q_1 = 16$ and Q_2 ranging from 18 to 40 with intervals of 2.

For CBR Intra-frame transcoding, 150 frames (each 352 × 240) of the sequences *Flower Garden* and *Football* were used. The sequences are initially encoded at a high quality with a bit-rate of 8Mb/s. Table 5.4 shows the average PSNR for different transcoding output bit rates. Two cases are shown: transcoding using selective requantization (PSNR_S) and normal transcoding (PSNR_N). Here, Δ is the difference between PSNR_S and PSNR_N. In normal transcoding, the value Q_2 determined by the rate control algorithm is directly used for requantization. Notice the improvement achieved using selective requantization. For example, the increase in PSNR of *Flower Garden* is 1.8 dB at a transcoding output bit rate of 7Mb/s, and 1.3 dB at a bit rate of 6Mb/s. Note that the performance of this method differs for different transcoding bit rates. For CBR transcoding of Intra and Inter frames, Table 5.5 shows the PSNR of normal transcoding and of transcoding using selective requantization for five sequences: *Flower Garden* (Flr.), *Football* (Ftb.), *Table Tennis* (Ten.), *Miss America* (Mis.), and *Salesman* (Slm.) with a group of pictures (GOP) structure N=15, M=3. Transcoding input and output bit rates are respectively 2Mb/s and 1.5Mb/s. It can be seen that selective requantization outperforms normal transcoding.

In summary, results show that selective requantization improve the quality of the transcoded sequence. The proposed method is more advantageous for Intra-frame transcoding.

5.4 Combining Both Requantization Techniques

In general, both requantization methods presented in this chapter are best suited for Intra-frame transcoding. In terms of complexity, selective requantization is simpler than the Laplacian parameter estimation method. The advantage of using one

Table 5.4: Average PSNR for Intra-frame transcoding of *Flower Garden* and *Football*. Selective requantization ($PSNR_S$) versus normal transcoding ($PSNR_N$). The input bit rate is 8Mb/s.

Bit Rate	$PSNR_S$ (dB)	$PSNR_N$ (dB)	Δ (dB)
Flower Garden			
7	29.1	27.3	1.8
6	28.0	26.7	1.3
5	27.3	26.6	0.7
4	26.6	26.2	0.4
Football			
7	36.0	34.1	1.9
6	35.1	33.6	1.5
5	34.4	33.5	0.9
4	33.7	33.4	0.3

Table 5.5: Average PSNR for transcoding of five sequences with a GOP structures N=15, M=3. Selective requantization ($PSNR_S$) versus normal transcoding ($PSNR_N$).

Transcoding PSNR (dB)					
Case	Flr.	Ftb.	Ten.	Slm.	Mis.
$PSNR_S$	25.9	29.8	33.9	38.6	41.5
$PSNR_N$	25.8	29.6	33.4	38.0	40.9
Δ	0.1	0.2	0.5	0.6	0.6

Table 5.6: Average PSNR for Intra-frame transcoding of *Flower Garden* and *Football*. Combined requantization methods (PSNR_{LS}) versus normal transcoding (PSNR_N). The input bit rate is 8Mb/s.

Bit Rate	PSNR_{LS} (dB)	PSNR_N (dB)	Δ (dB)
Flower Garden			
7	29.7	27.3	2.4
6	28.1	26.7	1.4
5	27.3	26.6	0.7
4	26.6	26.2	0.4
Football			
7	36.3	34.1	2.2
6	35.3	33.6	1.7
5	34.6	33.5	1.1
4	33.8	33.4	0.4

method over the other may depend on various factors such as the nature of the video and the GOP coding structure. For instance, because the estimated parameters from the last frame are used for the next frame of the same type, longer GOP structures, rapid scene changes and scene cuts may affect the performance of the Laplacian parameter estimation method. However, combining both methods will improve the performance of requantization especially in the above situations.

In this section, selective requantization is combined with the Laplacian parameter estimation method and results are shown in Table 5.6 for CBR Intra-frame transcoding. Note that combining both requantization techniques results in a better PSNR than in each individual method. While the improvement of the combined requantization techniques is small, it may vary for different sequences.

5.5 Summary of Chapter 5

In this chapter, two requantization methods for transcoding of MPEG compressed video were proposed. These methods use information available in the compressed video to achieve a higher transcoding quality. The first method, assumes Laplacian distributions for the original DCT coefficients. A Laplacian parameter for each coefficient is estimated at the transcoder from the quantized input DCT coefficients. This information is then used to reconstruct the input DCT coefficients before requantization. The second method, selective requantization, is based on avoiding critical ratios of the two cascaded quantizations (encoding versus transcoding) that either lead to larger transcoding errors or require a higher bit budget. Both presented methods are simple to implement and do not require side information. The chapter also presented experimental results that illustrate the improvement achieved in the transcoding quality by using the above requantization methods.

Chapter 6

Joint Transcoding of Multiple MPEG Video Bitstreams

6.1 Multi-program Video Transmission

In a multi-program video distribution environment, several video sequences are transmitted over a single communication channel. In situations where the communication channel has a lower capacity than that required by the video bitstreams, transcoding of these sequences may be necessary. A straightforward approach for transcoding these sequences would be to transcode each sequence independently at a constant bit-rate (CBR) such that their combined bit-rates meet the channel capacity. However, since scene complexities of different sequences usually vary with time, variable bit-rate (VBR) transcoding with joint bit allocation to meet the channel bandwidth is more efficient in allocating the available bits among sequences according to their scene complexities [13]. Consequently, joint transcoding reduces the variation in picture quality between the video sequences and within the individual sequences, as compared to independent transcoding.

6.2 Joint Transcoding

Many researchers addressed the problem of transmitting multiple VBR video over a shared communications channel [9, 81, 82]. In [9], a joint bit allocation strategy for joint coding of multiple video programs is presented. It is shown that joint encoding results in a uniform picture quality among programs as well as within a program. The paper also discusses further constraints on bit allocation to prevent the encoder and decoder buffers from overflowing and underflowing. In [81], a method to transmit multiplexed VBR video to improve video quality and/or transmission capacity is presented. The proposed method uses a multichannel rate control algorithm that allows statistical multiplexing of VBR video to regulate encoders during buffer congestion periods. The joint rate control algorithm is engaged when channel bandwidth overflow is predicted. The available bandwidth is then distributed among the video programs depending on their relative complexity. In [82], lossless aggregation, a scheme for the delivery of VBR video streams from a video server to a group of users over a shared CBR channel is presented. The transmission schedules of the individual video streams are determined dynamically depending on their relative traffic characteristics. It is shown that the aggregation method achieves buffer reduction compared to independent transmission of individual video streams over separate channels.

This chapter presents a joint transcoder for transcoding multiple MPEG bitstreams simultaneously. Joint transcoding is a part of the contributions of this thesis [13]. It is shown that joint transcoding reduces the variation in picture quality between the video sequences as well as within each sequence, compared to independent transcoding. Figures 6.1 and 6.2 show respectively independent transcoding and joint transcoding of N video sequences, as described in this work. In contrast to independent transcoding where each rate control works independently, in joint transcoding, all transcoders share the same rate control. The joint rate control uses updated information every frame to distribute the bits between the sequences depending on their relative coding complexities. Once the number of bits for a sequence frame is determined, a virtual buffer similar to that of the test model document version 5 ($TM5$) [39] is used in the transcoder to distribute the bits between the macroblocks of this frame. Note that in joint transcoding, each transcoder produces a variable bit-rate stream. Multiplexing the resulting N bitstreams yields a constant bit-rate stream that meets the channel bandwidth.

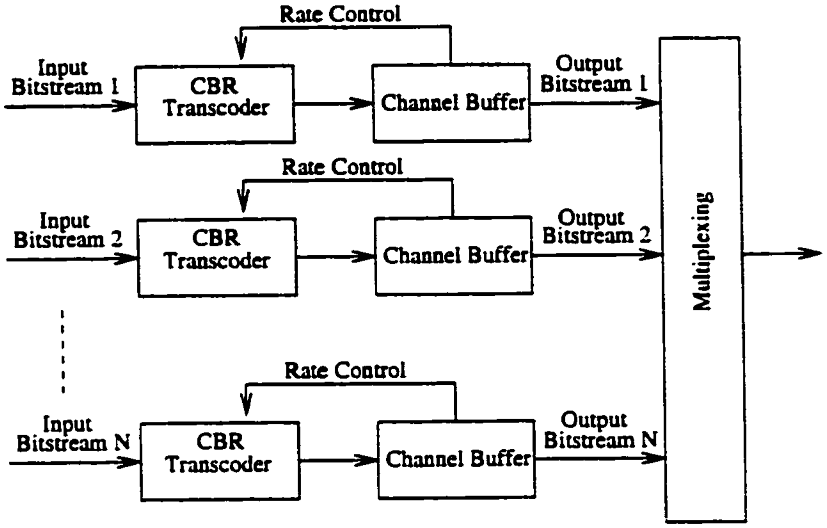


Figure 6.1: Independent transcoding of multiple video bitstreams.

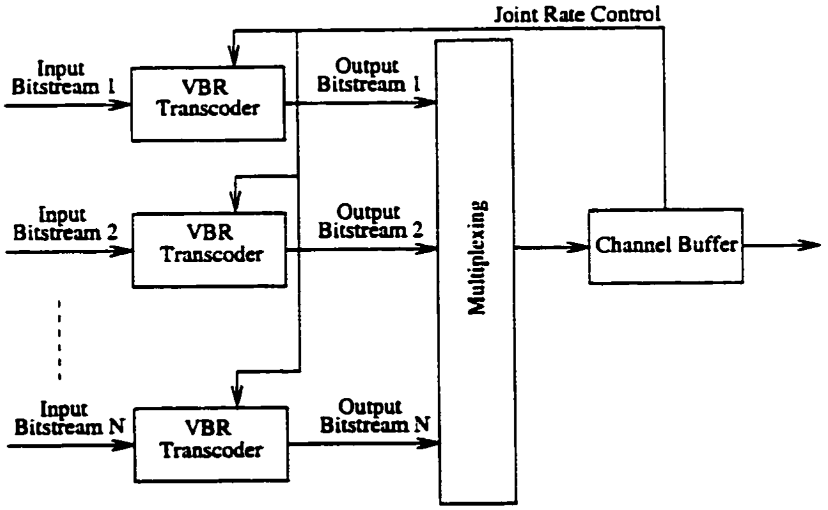


Figure 6.2: Joint transcoding of multiple video bitstreams.

6.3 Joint Bit-Allocation

In [83], a bit allocation for joint encoding of multiple video programs was presented. In this thesis, the joint bit-allocation algorithm used in joint transcoding is implemented as described in [83] with a few modifications. In [83], a binary tree search algorithm was used to determine a global quantization parameter for all program frames at any given time instant. This required five trials, each involved quantization and variable-length coding of every program frame, in order to choose a global quantization parameter that results in a bitrate closest to the target bitrate. The complexity associated with this approach will impose extra delays in transcoding and can be crucial especially for real-time applications. Furthermore, controlling the target bitrate for transcoding using a global quantization parameter may be hard to achieve. In this thesis, the joint bit-allocation algorithm presented in [83] has been modified such that bit allocation at the program frame level is performed similar to the test model document version 5 ($TM5$) [39] where a virtual buffer is used in transcoding to distribute the bits between the macroblocks of a program frame. An average quantization parameter is then computed after coding the frame and is used in determining the coding complexity of the next program frame of the same picture-type. Furthermore, the joint bit-allocation algorithm presented in [83] has been modified to include the activity measure (Equation 4.4) for requantization of macroblocks. The activity reflects the relative complexity of different macroblocks within a program frame. Since the Human Visual System is more sensitive to coding errors in the low-frequency regions of the picture (i.e., low-activity regions), these regions are usually requantized with a finer quantization step size. This section illustrates the joint bit-allocation algorithm used in joint transcoding.

Assuming joint transcoding of N video bitstreams, define a super frame as a group of N frames, one frame from each sequence, taken at the same time instant. The super frames are organized into super GOPs (group of pictures) where each super GOP contains M super frames, with M given by

$$M = LCM(M_1, M_2, \ldots, M_N) \tag{6.1}$$

where M_n, $n = 1, 2, \ldots, N$ is the GOP length of sequence n and LCM denotes the least common multiple operation. The above equation ensures that all super GOPs are identical and contain the same number of I, P, and B pictures. Thus, each super GOP contains $N \times M$ frames and may be assigned a target number of bits G given by

$$G = N \times M \times \frac{bit\_rate}{frame\_rate} \tag{6.2}$$

6.3.1 Bit Allocation at the Super Frame Level

Let T_m be the target number of bits for a super frame m and let $T(n, m, t)$ be the target bits for frame m of sequence n and with picture type t. Moreover, let $S(n, m, t)$ be the number of bits generated by coding this frame. Clearly, the sum of $T(n, m, t)$ over all sequences in super frame m should equal T_m. Furthermore, in order to control the bit rate, $S(n, m, t)$ should be close to $T(n, m, t)$. Thus, to derive the target number of bits for a sequence frame, it is assumed that $S(n, m, t)$ equals $T(n, m, t)$. Thus,

$$\sum_{n=1}^{N} T(n, m, t) = T_m \tag{6.3}$$

and

$$S(n, m, t) = T(n, m, t) \tag{6.4}$$

Moreover, the sum of T_m over a super GOP should satisfy

$$\sum_{m=1}^{M} T_m = G \tag{6.5}$$

Similar to the test model TM5 [39], let $\mathcal{X}(n,m,t)$ be the global complexity measure of frame m of program n and with a picture type t. Thus, $\mathcal{X}(n,m,t)$ can be expressed as

$$\mathcal{X}(n,m,t) = S(n,m,t)Q(n,m,t) \tag{6.6}$$

where $S(n,m,t)$ is the number of bits generated by coding frame m of program n, and $Q(n,m,t)$ is the respective average quantization parameter used in coding this frame.

Ideally, it is desired a constant quality (PSNR) between the different video sequences as well as within each sequence. This maybe achieved by using a constant quantization parameter for all sequence macroblocks. However, it is difficult to control the bit-rate using a constant quantization parameter for all the macroblocks. To ease this, the average quantization parameter $Q(n,m,t)$ over the frame is considered. Thus, in order to achieve a constant quality between the video sequences, an equal average quantization parameter for all sequences in a super frame is desired, i.e.,

$$Q(1,m,t) = Q(2,m,t) \ldots = Q(i,m,t) \ldots = Q(N,m,t) \tag{6.7}$$

From Equation 6.6, this requirement translates into

$$\frac{S(1,m,t)}{\mathcal{X}(1,m,t)} = \frac{S(2,m,t)}{\mathcal{X}(2,m,t)} = \ldots = \frac{S(i,m,t)}{\mathcal{X}(i,m,t)} = \ldots = \frac{S(N,m,t)}{\mathcal{X}(N,m,t)} \tag{6.8}$$

It follows from Equations 6.3 and 6.4,

$$S(1,m,t) + S(2,m,t) + \ldots + S(i,m,t) + \ldots + S(N,m,t) = T_m \tag{6.9}$$

To obtain an equation for the target number of bits for sequence i in super frame m, the left-hand side of the above equation may be expressed in terms of S(i,m,t). Thus,

$$\frac{\mathcal{X}(1,m,t)}{\mathcal{X}(i,m,t)}S(i,m,t)+\ldots+S(i,m,t)+\ldots+\frac{\mathcal{X}(N,m,t)}{\mathcal{X}(i,m,t)}S(i,m,t)=T_m \quad (6.10)$$

From Equations 6.4 and 6.10, the target number of bits for sequence i in super frame m can be written as

$$T(i,m,t)=\frac{\frac{1}{K_{i,m}}\mathcal{X}(i,m,t)}{\sum_{n=1}^{N}\frac{1}{K_{n,m}}\mathcal{X}(n,m,t)}T_m \quad (6.11)$$

where similar to the test Model TM5, $K_{n,m}$ is a constant factor introduced to compensate for different picture types (I, P, or B). Clearly, $T(i,m,t)$ depends on the global complexities of all sequence frames in super frame m.

6.3.2 Bit Allocation at the Frame Level

Once a target bit is allocated for a frame m of sequence i, a buffer is used to distribute the bits on between the macroblocks of this frame. The buffer fullness is computed and is used to control the bit rate. The fullness of the buffer is measured each macroblock and the quantization stepsize is increased or decreased accordingly to prevent buffer overflows and underflows. Before encoding macroblock l of frame m, the buffer fullness $b_l(i,m,t)$ is computed as

$$b_l(i,m,t)=b_0(i,m,t)+B_{l-1}(i,m,t)-\frac{T(i,m,t)\times(l-1)}{L} \quad (6.12)$$

where $b_0(i,m,t)$ is the initial buffer fullness for this frame, $B_{l-1}(i,m,t)$ is the number of bits spent by encoding all macroblocks in this frame up to, and including, macroblock $l-1$, $T(i,m,t)$ is the target number of bits assigned to frame m, and L is the total number of macroblocks in this frame. After encoding all L macroblocks in a frame, the final buffer fullness value, $b_L(i,m,t)$, is used as the initial buffer fullness for the next frame of the same type.

6.3.3 Bit Allocation at the Super GOP Level

At the start of transcoding, the super GOP is assigned the target bits, G, given by Equation 6.2. Let T_G be the remaining number of bits for a super GOP. Before transcoding the first super frame in the next super GOP, T_G is updated as

$$T_G = T_G + G \tag{6.13}$$

Within a super GOP, and after coding a super frame $j-1$, T_G is updated as follows

$$T_G = T_G - \sum_{n=1}^{N} S(n, j-1, t) \tag{6.14}$$

The target number of bits for the next super frame in the super GOP is given by

$$T_j = \frac{\sum_{n=1}^{N} \frac{1}{K(t(n,j))} \mathcal{X}(n,j,t)}{\sum_{m=j}^{M} \sum_{n=1}^{N} \frac{1}{K(t(n,m))} \mathcal{X}(n,m,t)} T_G \tag{6.15}$$

Clearly, the target number of bits for super frame j is proportional to its complexity. Note that the above equation depends on the global complexities of current and future frames. This requirement is removed in the experiments by using the most recent complexity measures for each sequence and for each picture type.

Illustrative Experiments on joint transcoding are presented in the next section.

6.4 Experiments on Joint Transcoding

To illustrate the performance of joint transcoding of multiple MPEG bitstreams, 150 frames (each 352 × 240) of the sequences *Flower Garden*, *Football*, *Table Tennis*, *Salesman* and *Miss America* were used with a color sampling ratio of 4:2:0. The

Figure 6.3: Frame No. 15 of Flower Garden

Figure 6.4: Frame No. 15 of Football

GOP length is 15, with a structure IBBPBBPBBPBBPBB. The original sequences are shown in Figures 6.3– 6.7 .

Figures 6.8 and 6.9 show the PSNR (Y-component) versus frame number for independent transcoding and joint transcoding of the above sequences, respectively. Both transcoding cases use drift correction. Each sequence was originally coded at 2Mb/s. The total input and output bit rates for joint transcoding are 10Mb/s and 7.5Mb/s, respectively. In independent transcoding, each sequence is transcoded separately at a constant bit rate of 1.5Mb/s. Notice that joint transcoding reduces the quality (PSNR) variation between the video sequences, as compared to independent transcoding. Variation is reduced more at higher bit rates and higher transcoding

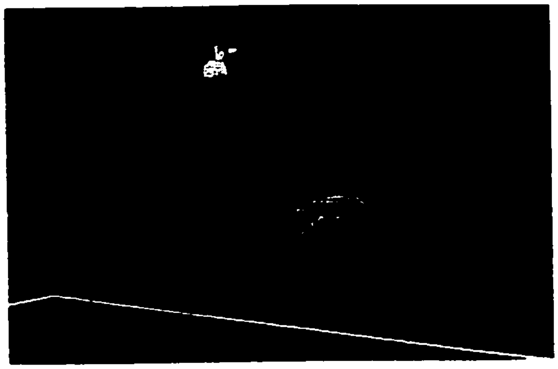


Figure 6.5: Frame No. 15 of Table Tennis

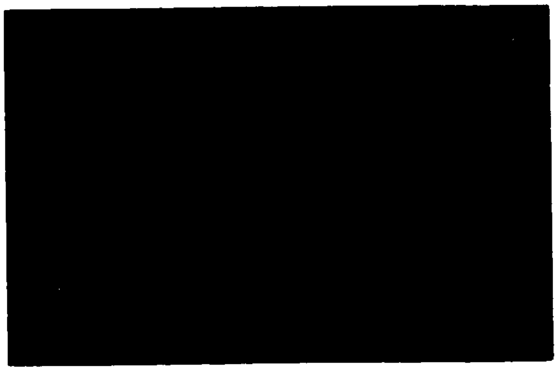


Figure 6.6: Frame No. 15 of Salesman

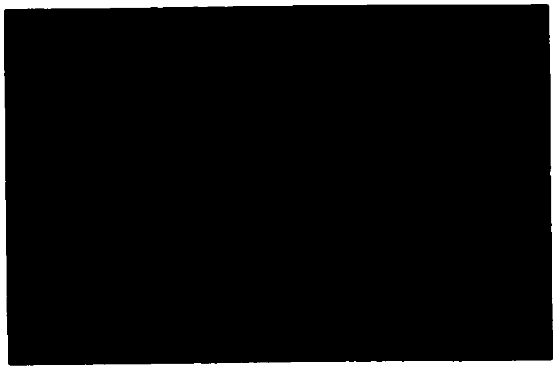


Figure 6.7: Frame No. 15 of Miss America

Table 6.1: PSNR of joint and independent transcoding. The total input and output bit rates of the five sequences are 10Mb/s and 7.5Mb/s, respectively.

Transcoding with drift correction								
Case	PSNR (dB)							
	Flr.	Ftb.	Ten.	Slm.	Mis.	Mean	Min	Var
Joint	28.7	32.5	34.3	36.2	40.0	34.3	25.6	16.9
Indep.	25.8	29.6	33.4	38.0	40.9	33.5	23.5	32.5
Δ	2.9	2.9	0.9	-1.8	-0.9	0.8	2.1	-15.6

Transcoding with no drift correction								
Case	PSNR (dB)							
	Flr.	Ftb.	Ten.	Slm.	Mis.	Mean	Min	Var
Joint	28.7	32.4	33.8	34.9	39.3	33.8	25.6	15.6
Indep.	24.9	29.0	32.7	37.2	40.0	32.8	22.4	33.9
Δ	3.8	3.4	1.1	-2.3	-0.7	1.0	3.2	-18.3

Table 6.2: Transcoding output bit rates.

Case	Bit Rate in Mb/s				
	Flr.	Ftb.	Ten.	Slm.	Mis.
Joint (drift correct.)	2.0	2.0	1.8	1.1	0.6
Joint (no drift correct.)	2.0	2.0	1.8	1.0	0.7
Independent	1.5	1.5	1.5	1.5	1.5

ratios.

Table 6.1 shows the average PSNR (Y-component) in both independent and joint transcoding of the above sequences. Results are shown for transcoding with drift correction and with no drift correction. Here, Δ denotes the difference in PSNR between joint and independent transcoding. The bit rates for both joint and independent transcoding are shown in Table 6.2.

Note that joint transcoding results in an increase in PSNR of higher complexity sequences, as compared to independent transcoding. For example, the PSNR of *Flower Garden* is increased by 2.9 dB and 3.8 dB for joint transcoding with drift correction and joint transcoding with no drift correction, respectively. Furthermore, the average gain achieved by using drift correction is 0.5 dB and 0.7 dB for joint

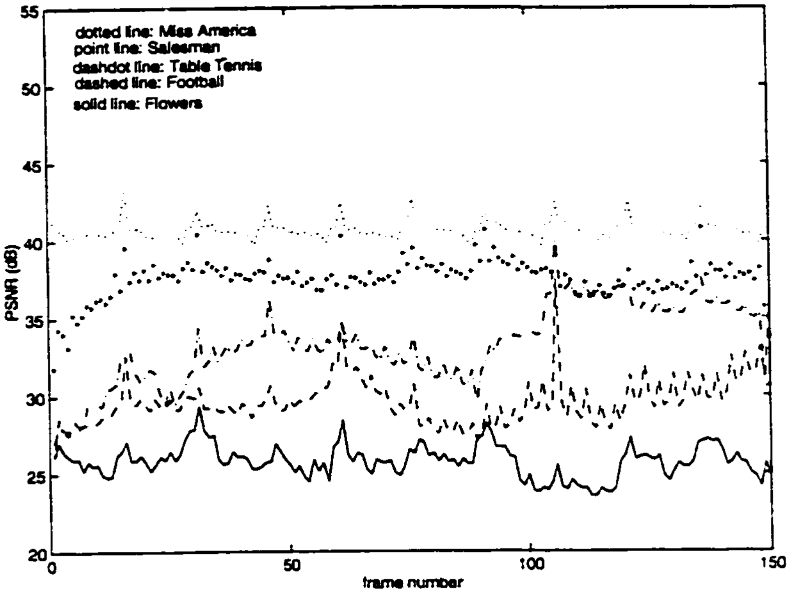


Figure 6.8: Independent transcoding (with drift correction) of five bitstreams. The input and output bit rates for each bitstream are 2Mb/s and 1.5Mb/s, respectively.

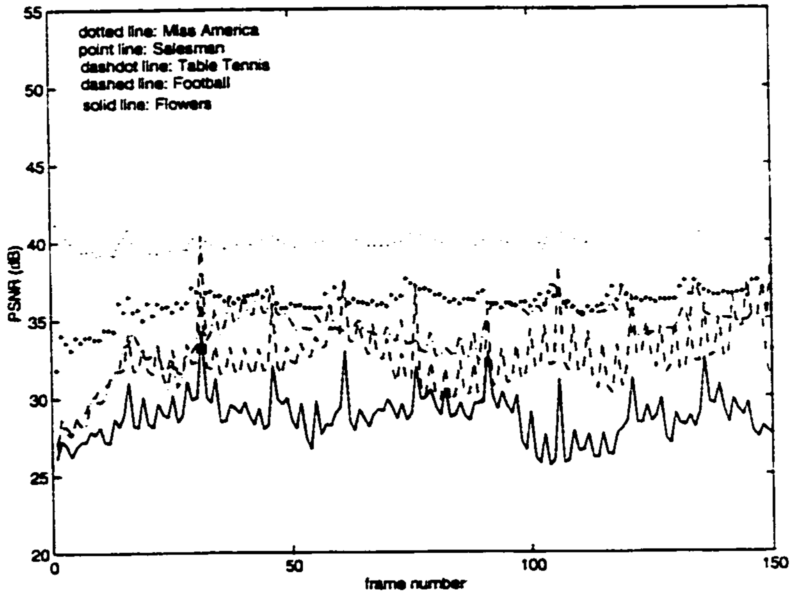


Figure 6.9: Joint transcoding (with drift correction) of five bitstreams. The total input and output bit rates are 10Mb/s and 7.5Mb/s, respectively.

transcoding and independent transcoding, respectively. In general, this gain is higher for larger transcoding ratios.

Notice that the average gain achieved by using joint transcoding with no drift correction is higher than independent transcoding with drift correction. This may be important in real-time and low delay applications where joint transcoding with no drift correction can be used and still provides a better average PSNR than independent transcoding with drift correction.

6.5 Minimizing the Quality-Variation between the Jointly Transcoded sequences

In multi-program video transmission such as video on demand (VOD) services and digital TV broadcasting, an important concern for a video service provider may be to maintain a certain minimum picture quality as well as a constant quality between the video programs. In order to minimize the quality-variation between the jointly transcoded sequences and to achieve a better minimum picture quality [12], the picture complexity measure used in this section is different than the complexity measure given by Equation 6.6. The presented method uses a picture-complexity measure based on the actual coding distortion in the transcoded frames, then allocates accordingly the available bits to explicitly reduce the variation in quality between the transcoded sequences.

In this work, the PSNR is used as an objective measure of the picture quality. However, the idea presented in this section can be generalized, i.e., any other objective video quality metric that may capture the subjective information in the video could be substituted for the PSNR. The performance of joint transcoding using the new picture complexity measure is compared to independent transcoding of the sequences at constant bit rates and to transcoding with a joint bit-allocation scheme that uses a $TM5$-like picture-complexity measure. Results show that the proposed bit-allocation method is superior in terms of minimizing the quality variation between the video sequences and within the individual sequences, while maintaining a better minimum picture quality than the other transcoding schemes presented in this section.

Using a similar concept of super frame and super GOP discussed in Section 6.3, consider joint transcoding of N video sequences. let $\mathcal{X}(n,m,t)$ be the global complexity measure of frame m of sequence n and with picture type t. In a $TM5$ topology, $\mathcal{X}(n,m,t)$ is defined as the product $S(n,m,t)Q(n,m,t)$ where $Q(n,m,t)$ is the average quantization parameter used in coding frame m of sequence n and with picture type t (Equation 6.6). However, in this section, $\mathcal{X}(n,m,t)$ is defined as

$$\mathcal{X}(n,m,t) = S(n,m,t)e(n,m,t) \tag{6.16}$$

where $e(n,m,t)$ is the root mean-square error (RMSE), or quantization error, used as an objective measure of the transcoding quality of this picture and defined as

$$e(n,m,t) = \sqrt{\frac{1}{64L}\sum_{l=1}^{L}\sum_{u=0}^{7}\sum_{v=0}^{7}\left[X^{l}_{n,m,t}(u,v) - Y^{l}_{n,m,t}(u,v)\right]^2} \tag{6.17}$$

where L is the number of 8×8 DCT blocks in the picture, $X^{l}_{n,m,t}(u,v)$ is the inverse quantized coefficient of the lth DCT block at the transcoder's input, and $Y^{l}_{n,m,t}(u,v)$ is the reconstructed DCT coefficient after requantization. Note that Equation 6.17 can be applied to both Intra-frames and Inter-frames, as the error in P and B pictures is only due to the quantization of the residue. However, in the case of P and B pictures, a compensation for drift errors that result from requantizing a motion-compensated compressed video has to be added to the input DCT coefficients $X_{n,m,t}(u,v)$ prior to requantization.

Ideally, in order to minimize the quality variation (PSNR) between the different video sequences, it is desired equal quantization errors for all sequences in a super frame, i.e.,

$$e(1, m, t) = e(2, m, t) \ldots = e(N, m, t) \tag{6.18}$$

Similar to the analysis given in Section 6.3, the target number of bits at the frame level and at the super frame level are given by Equations 6.11 and 6.15, respectively. Clearly, the target number of bits $T(i, m, t)$ for a program i depends on the global complexities of all sequence frames in super frame m, or alternatively, $T(i, m, t)$ depends on $e(n, m, t)$, i.e., the quantization errors of all sequences in this super frame.

This experiment was performed using 150 frames (each 352 × 240) of the sequences *Football* (Ftb.), *Table Tennis* (Ten.), *Salesman* (Slm.) and *Miss America* (Mis.). Only Intra-frame coding was used and a total coding bit rate of 20 Mb/s. In independent transcoding, the $TM5$ rate control algorithm was used and each individual sequence was transcoded at a fixed bit rate of 5 Mb/s. Figures 6.10, 6.11 and 6.12 show respectively the PSNR versus frame number of independent transcoding, joint transcoding using a $TM5$-like global picture complexity, and joint transcoding using the proposed global picture complexity given by Equation 6.16. The corresponding number of coding bits per frame are shown in Figures 6.13, 6.14 and 6.15, respectively.

Notice from the figures that both joint transcoding methods reduce the quality variation between the video sequences, when compared to independent transcoding. However, quality variation is minimized when using the proposed complexity measure of Equation 6.16.

Figures 6.16, 6.17, and 6.18 show the total bits per frame unit (or 1/30 of a second) after multiplexing the transcoded sequences. The Minimum and maximum bits per frame are shown on each plot. The average total bitrate for the three above cases are close. The fluctuations of the total bit rate above and below the average are respectively in the range of ±4%, ±5%, and ±5% for independent transcoding, joint transcoding using a $TM5$-like global picture complexity, and for joint transcoding using the proposed global picture complexity. It can be seen that the fluctuation in the total bit rate of the multiplexed transcoded sequences is comparable for the above three cases.

Tables 6.3 and 6.4 show respectively the average PSNR and bit rates of independent transcoding (Indep.), joint transcoding with a $TM5$-like picture complexity (Joint$_A$) and joint transcoding using the new picture complexity given by Equation 6.16 (Joint$_B$). The first thirty frames were not included in the PSNR calculations. Furthermore, as it is more important to maintain a minimum picture quality rather than an average picture quality, the minimum PSNR ($PSNR_{min}$) for all sequences was computed. Furthermore, the variance of PSNR values ($PSNR_{var}$) was calculated to demonstrate how the proposed method minimizes the quality variation between the frames of one sequence and among the sequences (smaller variance means less quality variation). For N sequences of L frames each, $PSNR_{min}$ and $PSNR_{var}$ are given by

$$PSNR_{min} = \min\{PSNR(n,l)\} \qquad \forall n, l \tag{6.19}$$

and

$$PSNR_{var} = \frac{1}{NL}\sum_{n=1}^{N}\sum_{l=1}^{L}\left(PSNR(n,l) - \overline{PSNR}\right)^2 \tag{6.20}$$

where $\overline{PSNR}$ is the average PSNR over the $N \times L$ frames.

The best scenario in coding of multiple video programs may be one that

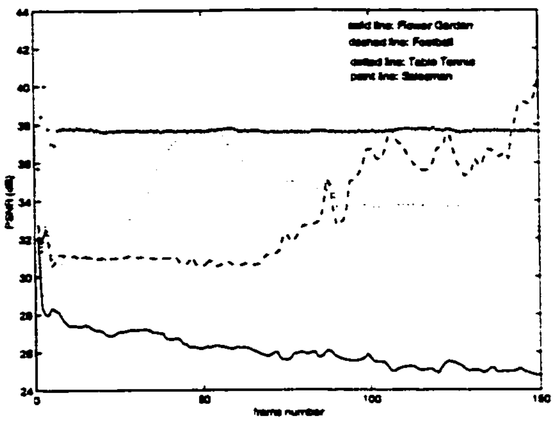


Figure 6.10: Independent transcoding of four sequences.

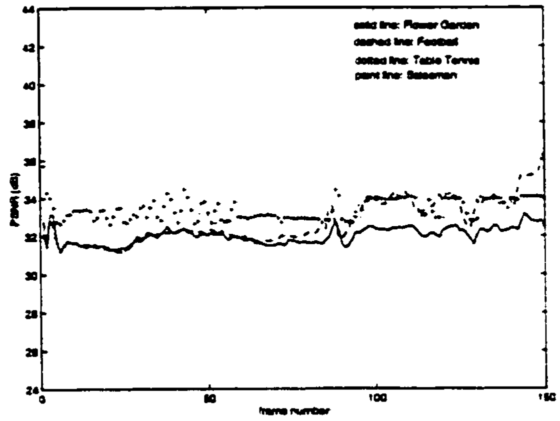


Figure 6.11: Joint transcoding of four sequences using a $TM5$-like picture complexity.

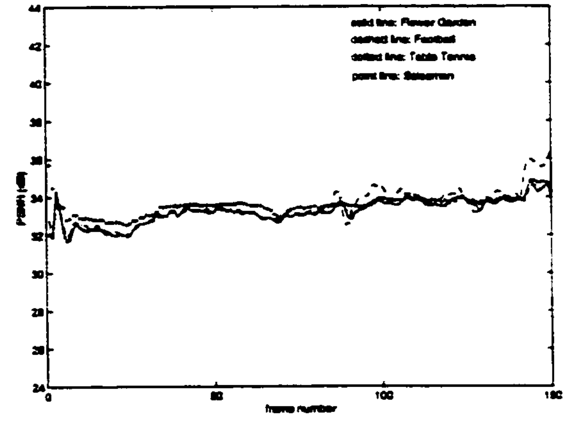


Figure 6.12: Joint transcoding of four sequences using the proposed method.

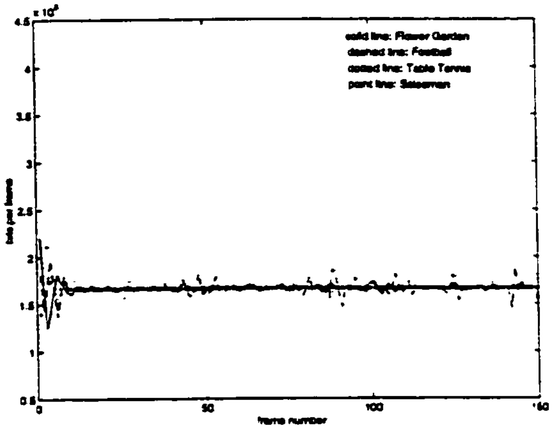


Figure 6.13: Number of bits per frame for independent transcoding (Figure 6.10).

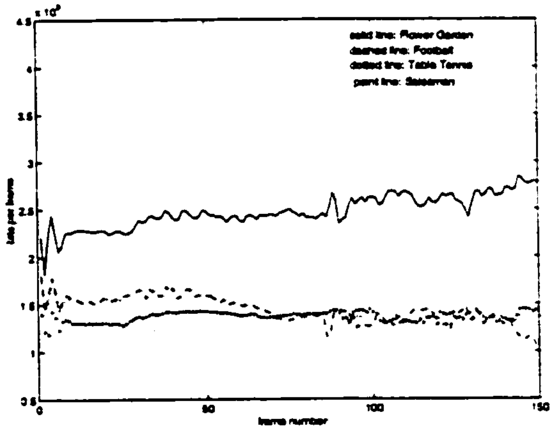


Figure 6.14: Number of bits per frame for joint transcoding using a $TM5$-like picture complexity (Figure 6.11).

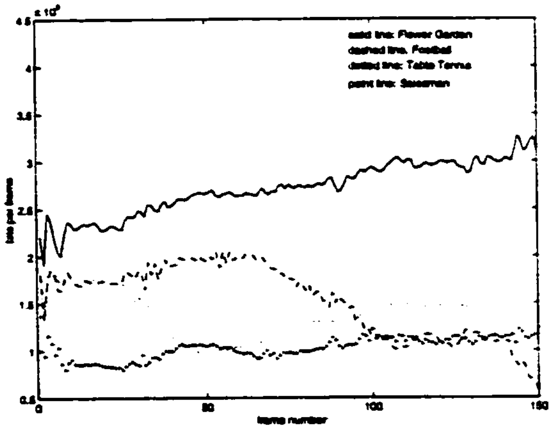


Figure 6.15: Number of bits per frame for joint transcoding using the proposed picture complexity method (Figure 6.12).

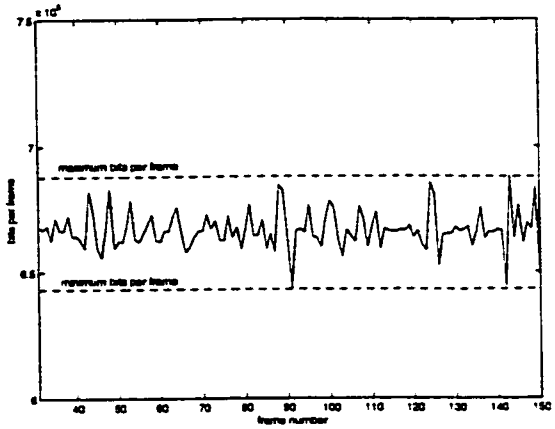


Figure 6.16: Total bits per frame of the multiplexed sequences for independent transcoding (Figure 6.10).

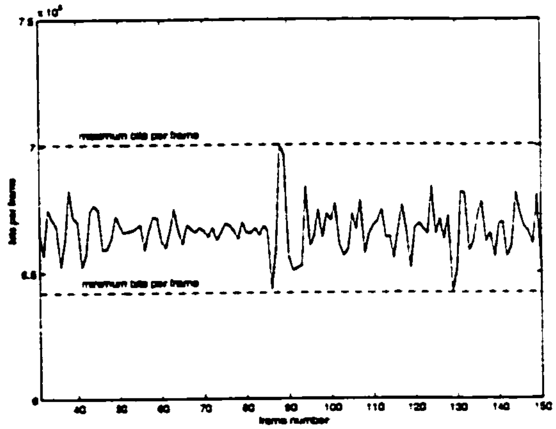


Figure 6.17: Total bits per frame of the multiplexed sequences for joint transcoding using a $TM5$-like picture complexity (Figure 6.11).

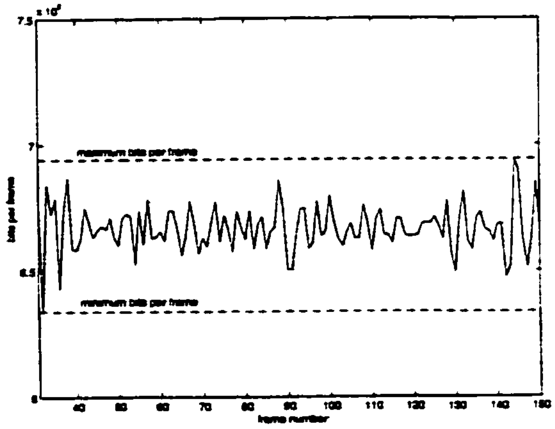


Figure 6.18: Total bits per frame of the multiplexed sequences for joint transcoding using the proposed picture complexity method (Figure 6.12).

achieves the smallest $PSNR_{var}$ and the maximum $PSNR_{min}$ and $\overline{PSNR}$. Although it may not be always possible to minimize the PSNR variance ($PSNR_{var}$) and maximize the average PSNR ($\overline{PSNR}$), an important concern for a video service provider may be to maintain a certain minimum picture quality ($PSNR_{min}$) as well as a constant quality between the video programs (minimum $PSNR_{var}$). Following this concept, it can be seen from Table 6.3 that the proposed picture complexity method has the smallest variance and the highest $PSNR_{min}$ (the notations Mean, Min, and Var in Table 6.3 correspond to $\overline{PSNR}$, $PSNR_{min}$, and $PSNR_{var}$, respectively). Notice also that the proposed method has the highest average PSNR, ($\overline{PSNR}$), when compared to the other presented methods.

Table 6.3: Transcoding average PSNR for four sequences (Intra-frame coding). The total coding bit rate is 20Mb/s.

Coding	PSNR (dB)						
	Flr.	Ftb.	ten.	Slm.	Mean	Min	Var
Indep.	25.7	33.9	34.7	37.6	33.0	24.7	22.3
Joint$_A$	32.1	33.0	32.9	33.4	32.9	31.4	0.7
Joint$_B$	33.5	33.7	33.6	33.7	33.6	32.5	0.3

Table 6.4: Encoding bit rates of the sequences in Table 6.3.

Coding	Bit Rate in Mb/s				
	Flr.	Ftb.	Ten.	Slm.	Total
Indep.	5	5	5	5	20
Joint$_A$	7.4	4.3	4.2	4.1	20
Joint$_B$	8.1	4.6	4.2	3.1	20

6.6 Interaction of the Joint Transcoder with Channel Traffic

This section studies the interaction of the joint transcoder with channel traffic when multiple MPEG video sequences are transmitted over a fixed communication channel that is partially occupied by another MPEG traffic. In the experiments, the test sequences *Flowers* (Flr.), *Football* (Ftb.), *Table Tennis* (Ten.), *Salesman* (Slm.) and *Miss America* (Mis.) were used. Each of these sequences has 150 frames (each 352 × 240) and contains scenes with different motion and texture. In order to simulate a variable bit-rate (VBR) video traffic, 30 frames were taken from each of the above sequences and concatenate them to form a new sequence of 150 frames. This sequence is then MPEG encoded at a fixed quantizer scale. The resulting number of bits per frame are scaled down to any desired average bit rate and is used to represent the channel traffic. Figure 6.19 illustrates the traffic profile in bits per frame for an average bit rate of 2Mb/s.

6.6.1 Experiment 1

In the first experiment, the five sequences are initially encoded each at 2Mb/s, with a total bit rate of 10Mb/s. It is assumed that these sequences are transmitted over a 10Mb/s channel bandwidth. The channel also carries a VBR video traffic. The transcoder is placed at channel node to perform bit-rate conversion on the encoded sequences such that the total bit-rate of transcoded sequences and traffic meet the channel bandwidth. The transcoder receive information about the traffic at the frame level. The experiment is repeated using traffic with different average bit-rate.

The performance of joint transcoding [13] was compared to independent

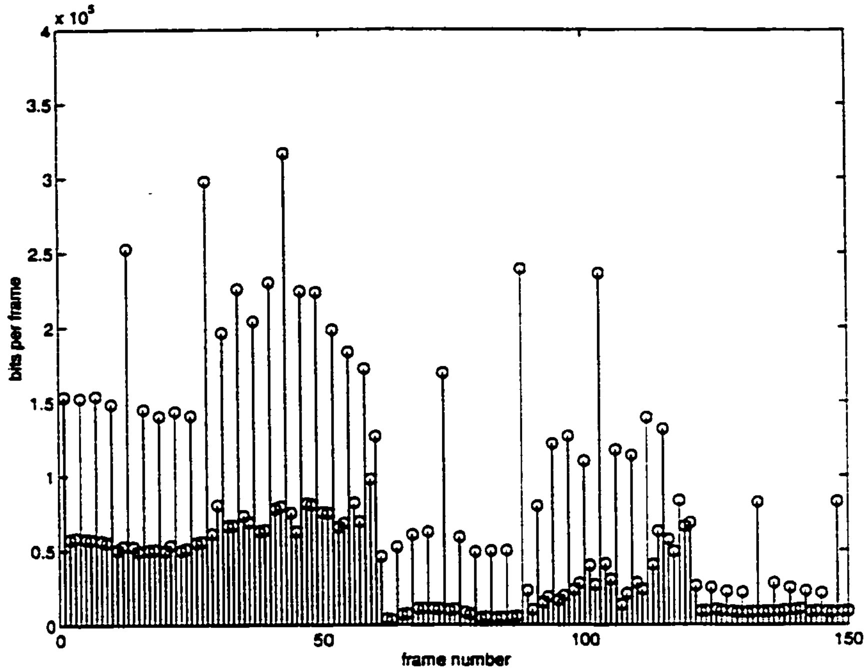


Figure 6.19: Real video traffic profile. Traffic is composed of the sequences Ftb., Flr., Slm., Ten., and Mis. in the same order (30 frames from each sequence).

transcoding, in the presence of channel traffic. The bit-rate control in transcoding is based on the Test Model document, version 5 ($TM5$) [39]. Tables 6.5 and 6.6 show respectively the average PSNR for joint transcoding and independent transcoding of the sequences for various average bit-rate traffic (shown in the first column). The transcoding bit-rates are shown in the second column. Moreover, the average PSNR (Mean), the minimum PSNR (Min), and the standard deviation (Std) were evaluated over all sequence frames.

As discussed earlier in this chapter, The best scenario in transcoding of multiple video programs may be one that achieves the smallest variation between the pictures quality (Std) while maintaining a higher minimum picture quality (Min) and average picture quality (Mean). Thus, it can be seen from Tables 6.5 and 6.6 that joint transcoding outperforms independent transcoding, even in the presence

Table 6.5: Joint transcoding of five sequences in the presence of channel traffic. The sequences are initially coded each at 2Mb/s. The channel has 10Mb/s bandwidth and carries VBR video traffic.

Traffic	Coding	Joint transcoding							
		PSNR (dB)							
(Mb/s)	(Mb/s)	Flr.	Ftb.	Ten.	Slm.	Mis.	Mean	Min	Std
0.0	10	28.5	32.4	35.9	40.6	42.4	36.0	25.5	5.5
1.0	9.0	28.2	31.8	35.2	39.9	41.5	35.3	25.4	5.3
2.0	8.0	27.8	31.0	34.7	39.4	41.4	34.9	24.8	5.6
3.0	7.0	27.2	30.2	34.0	38.5	40.9	34.2	23.1	5.7
4.0	6.0	26.4	29.5	33.1	37.2	40.2	33.3	22.4	5.7
5.0	5.0	25.4	28.7	32.2	35.8	39.8	32.4	22.1	5.9

Table 6.6: Independent transcoding of five sequences in the presence of channel traffic. The sequences are initially coded each at 2Mb/s. The channel has 10Mb/s bandwidth and carries VBR video traffic.

Traffic	Coding	Independent transcoding							
		PSNR (dB)							
(Mb/s)	(Mb/s)	Flr.	Ftb.	Ten.	Slm.	Mis.	Mean	Min	Std
0.0	10	28.5	32.4	35.9	40.6	42.4	36.0	25.5	5.5
1.0	9.0	27.7	31.3	35.2	40.2	42.1	35.3	24.7	5.8
2.0	8.0	27.1	30.7	34.6	39.5	41.9	34.7	23.7	6.0
3.0	7.0	26.2	30.0	33.9	38.6	41.6	34.1	22.5	6.2
4.0	6.0	25.2	29.2	33.0	37.6	40.9	33.2	22.1	6.3
5.0	5.0	24.0	28.3	32.2	36.5	40.4	32.3	21.8	6.5

of channel traffic.

6.6.2 Experiment 2

In this experiment, the sequences of Section 6.6.1 were used. The sequences are originally coded each at 2Mb/s. The purpose of this experiment is to find the effect of the traffic variance on the performance of the transcoder. The same traffic characteristics is used for all simulations. However, in each simulation, the variance of the traffic is changed by multiplying the traffic by a constant. In order to keep the target bit rate for transcoding the same throughout the experiment, the total allowable bandwidth for both traffic and video varies on average with traffic such that the average bit-rate for transcoding is kept constant. As in Section 6.6.1, the average PSNR, the minimum PSNR, and the standard deviation of the PSNR were evaluated for all sequence frames.

Table 6.7 shows the PSNR results for an average coding bit rates of 7.5 Mb/s and various average bit-rate traffic (shown in first column). The variance of the traffic is shown in the second column. Notice that the performance of joint transcoding improves as the traffic variance decreases. This was expected as the transcoder has more flexibility in allocating the bits between sequences and within the individual sequences.

Table 6.7: Transcoder's performance different traffic characteristics. Joint transcoding of five sequences initially coded each at 2Mb/s. The output bit-rate for transcoding is 7.5 Mb/s.

Traffic		Joint transcoding							
		PSNR (dB)							
Mean (bits/frame)	Var	Flr.	Ftb.	Ten.	Slm.	Mis.	Mean	Min	Std
1.0019e+05	9.9836e+09	27.2	30.2	34.0	38.5	40.9	34.2	23.1	5.7
1.3359e+05	1.7749e+10	26.9	30.1	34.0	38.5	40.9	34.1	22.5	5.9
1.6699e+05	2.7732e+10	26.7	29.9	33.9	38.3	40.9	34.0	22.3	6.0
2.0038e+05	3.9934e+10	26.6	29.7	33.7	38.0	40.9	33.8	22.3	6.0

6.7 Summary of Chapter 6

This chapter discussed the problem of multi-program video transmission over a fixed communication channel and presented a joint transcoder for transcoding multiple MPEG video bitstreams simultaneously. The joint bit allocation algorithm used in transcoding was further discussed and formulated. Results showed that the joint transcoder increases the PSNR of higher complexity sequences, and results in an overall gain in average PSNR and minimum PSNR when compared to independent transcoding of each sequence at a constant bit-rate. Joint transcoding also results in a smaller PSNR variance (minimize the variation in quality) as compared to independent transcoding.

The chapter also presented a joint bit-allocation method to implicitly minimize the quality variation (PSNR) between the jointly transcoded sequences. The scheme used a picture-complexity measure based on the actual coding errors in transcoded frames. The presented method was compared to joint transcoding using a $TM5$-like picture-complexity measure, and to independent transcoding. Results showed that the proposed method is superior in terms of minimizing the variation in quality between the video sequences as well as within the individual sequences. It also provides a better minimum quality (PSNR) within the transcoded sequences.

Finally, the interaction of the joint transcoder with channel traffic was studied. The performance of joint transcoding was compared to independent transcoding in the presence of channel traffic. Experiments showed that joint transcoding outperforms independent transcoding, even in the presence of channel traffic. However, the performance of joint transcoding improves as the traffic variance decreases.

Chapter 7

Conclusions and Future Directions

7.1 Remarks and Conclusions

With the increasing number of applications using compressed digital video for transmission over heterogeneous networks, transcoding is becoming an important issue for bitrate reduction to meet channels capacities. Transcoding provides flexibility in transmission bitrate especially if the bitrate is only known at transmission time. This flexibility is very important when the coded video is multicasted over channels with different capacities such as cable networks and satellite networks. Moreover, in situations such as network congestion, transcoding is a key technology for providing dynamic adaptation of the bitrate of compressed video to the available channel bandwidth.

Motivated by the importance of transcoding in current video technologies, this thesis investigated different issues related to transcoding. First, Chapter 3 addressed the degradations in picture quality which result from the repeated compression-decompression of video, known as multigeneration errors. These errors are introduced at each compression-decompression cycle, even if no bitrate reduction occurs between generations. Five mechanisms that cause continuous degradation in multigeneration of MPEG compressed video were identified, specifically, Pixel Domain Quantization (PDQ), Pixel Domain Clipping (PDC), Compression Control Parameters Variation (CCPV), Motion Vector Re-estimation (MVR) and Error Propagation due to Motion Compensation (EPMC). Experiments were presented to quantify the errors introduced by each of these mechanism. It was shown that PDQ and EPMC insert larger errors for higher bitrate video, while CCPV and MVR are more important at lower bitrates. The effect of PDC on multigeneration loss, for the sequences used in experiments, was very small.

The main focus of this thesis was on the problem of transcoding of MPEG compressed video. Two requantization methods were proposed to improve the quality of the transcoded video [7, 8]. The first method used Laplacian distributions to estimate the original DCT coefficients at the transcoder input [7]. A Laplacian parameter for each coefficient was estimated from the quantized input DCT coefficients. The Laplacian parameters were then used to reconstruct the input DCT coefficients at the local centroid of the distribution before requantization to reduce the errors in the rate conversion process. An algorithm was provided to estimate the Laplacian parameters at the transcoder for both midstep and midriser quantizers. Experiments showed that transcoding using the Laplacian pdf model outperforms normal transcoding, where the input DCT coefficients are uniformly reconstructed before requantization following the MPEG-2 uniform reconstruction equation. In VBR transcoding, the number of bits for the sequence used in Section 5.2.1 was reduced by a factor of 26% by using the proposed method. For CBR transcoding, the PSNR improvement for Intra-frame transcoding of *FlowerGarden* (Section 5.2.1) ranged from 2.4 to 0.3 dB depending on the output bitrate. In CBR Inter-frame transcoding, about 0.4 dB improvement was achieved for input and output bitrates of 2Mb/s and 1.5Mb/s, respectively.

Selective requantization [8], a second method for improving the transcoding of MPEG compressed video, was provided in this thesis. The proposed method is based on avoiding critical ratios between the two cascaded quantizations (encoding versus transcoding) that either lead to larger transcoding errors or require a higher bit budget. It was shown in Section 5.3 that the extra distortion introduced by two cascaded quantizations depends on the ratio between the stepsize of the two quantizers. For CBR Intra-frame transcoding, the improvement achieved using selective requantization (Section 5.3.1) ranged from 1.8 to 0.4 dB depending on the transcoding ratio. In CBR Inter-frame transcoding, the improvement was 0.4 dB, on average, for input and output bitrates of 2Mb/s and 1.5Mb/s, respectively.

Next, the thesis addressed the problem of multi-program video transmission over heterogeneous networks and provided a joint transcoder for transcoding multiple MPEG video bitstreams simultaneously [13]. In contrast to independent transcoding where each rate control works independently, in the presented technique a joint bit-allocation was used to distribute the bits between the sequences depending on their relative scenes complexities. It was shown that joint transcoding results in a more constant quality between the video sequences as well as within each sequence. This allowed for a better utilization of the available channel bandwidth. The joint bit-allocation algorithm was presented in details in Section 6.3. The experiments were conducted for two transcoding structures with different complexity and memory requirements: a trancoder with drift correction, and a transcoder that uses a direct requantization, i.e., with no drift compensation. A comparison between the output picture quality of joint transcoding versus independent transcoding was shown in Section 6.4. The PSNR improvement achieved in joint transcoding versus independent transcoding of five sequences was about 0.8 dB on average. For transcoding with drift correction. Moreover, joint transcoding achieved about 2.1 dB improvement in the minimum picture-quality measure. This measure is important for many video applications, such as video on demand (VOD) services, in maintaining an acceptable picture-quality level during a worst case transmission scenario.

Furthermore, Section 6.5 presented a joint bit-allocation method to implicitly minimize the quality variation between the jointly transcoded sequences [12]. In this method, a new picture complexity measure based on the actual coding distortion in the transcoded pictures was used. The method was compared to joint transcoding using a $TM5$-like picture-complexity measure. The new complexity measure method was tested for intra-frame joint transcoding of four sequences. The PSNR variance was reduced by 0.4 by using this method. Moreover, the method achieved an increase in the average PSNR and minimum PSNR of 0.7 dB and 1.1 dB, respectively.

Section 6.6 studied the interaction of the joint transcoder with channel traffic in the case of multi-program video transmission over a fixed communication channel. The Joint transcoder was tested in the presence of channel traffic. It was shown that joint transcoding outperforms independent transcoding and that this performance improves as the traffic variance decreases.

7.2 Future Directions

Chapter 3 addressed the problem of multigeneration and degradation in picture quality that results from the repeated compression and decompression of MPEG video. A challenging problem in multigeneration of MPEG video is watermarking, where hidden data such as owner's copyright protection and other security related issues are embedded in the compressed video. Many watermarking techniques use the visual masking effects based on the human visual system. In these techniques, a DCT coefficient can be modified to carry hidden information without causing visible changes in the picture. Future research may address robustness of video watermarking in a multigeneration environment. Chapter 3 showed that each compression/decompression cycle may alter the value of a DCT coefficient due to the interaction between the DCT-domain quantizer and the pixel-domain quantizer. This consequently may affect any watermarking information embedded in that DCT coefficient. It was shown however in Chapter 3 that an 8×8 DCT block of data will remain unchanged between generations if there exists a "trapping" pair in R^{64} between the DCT-domain and the pixel-domain quantizers. Thus, any watermark associated with that DCT block will survive multiple compression/decompression cycles. This is an important attribute of a successful watermarking scheme. Future research may include finding these trapping pairs and using their respective DCT blocks to do watermarking. Research also may investigate how much precision for data values in the pixel domain is needed to avoid the PDQ degradation mechanism.

Moreover, in a video distribution network, multigeneration of video typically involves image manipulation and editing. Future work may investigate the degradation caused by the compression/decompression cycles in a production manipulation environment. Multigeneration experiments may include image manipulation such as scene cuts, dissolves and fades, zooming, filtering, slow motion, screen splitting with two different video, and adding text to the original video.

The thesis mainly focused on the problem of transcoding of MPEG compressed video. Video transcoding is an important area of research and is increasingly finding wide varieties of applications. While this thesis addressed the specific transcoding problem of bitrate conversion of MPEG compressed video, this research is not limited to MPEG compression and can be extended to cover other compression standards such as JPEG and H.263. Moreover, other techniques such as frame dropping or changing the spatial resolution of the video may be combined to the requantization techniques used in this thesis to improve transcoding performance and to cover a wider area of applications. For example, a change in the spatial resolution of a pre-encoded video may be necessary in video transcoding in order to support receivers with different display capabilities.

Another transcoding problem is changing a compressed video format into another compressed format. For example, if a studio quality video is compressed and stored at a high bitrate using a small GOP structure or only intra-frame pictures for editing purposes, a change in picture type combined with a reduction in bitrate may be necessary for video distribution to end users.

For multi-program video transmission over a fixed communication channel, it was shown that joint transcoding provided a more constant quality between the sequences than independent transcoding. While Section 6.5 provided a technique to implicitly minimize the PSNR, it was observed in the experiments that low detailed sequences with large flat areas were subject to blocking artifacts when their bitrates were reduced implicitly to minimize the variation in PSNR between the sequences. This is because the PSNR may not always reflect the subjective quality of the video. Future research may include a modification of the picture complexity measure (Equation 6.16) to include an objective video-quality metric that captures subjective information in the video such as blocking or blurring artifacts. This measure can be used in transcoding to maintain a desired subjective quality between video programs and to reduce blocking artifacts.

REFERENCES

[1] A. T. Erdem and M. I. Sezan, "Multi-generation Characteristics of the MPEG Video Compression Standards," *IEEE Proceedings International Conference on Image Processing*, vol. 2, pp. 933–937, November 1994.

[2] C. Horne, T. Naveen, A. Tabatabai, R. O. Eifrig, and A. Luthra, "Study of the Characteristics of the MPEG2 4:2:2 Profile – Application of MPEG2 in Studio Environment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 251–272, June 1996.

[3] H. Sorial and W. E. Lynch, "Degradation Mechanisms in Multigeneration of MPEG Compressed Video," *Special issue on Visual Computing and Communications, Canadian Journal of Electrical and Computer Engineering*, vol. 23, pp. 5–9, January-April 1998.

[4] H. Sorial and W. E. Lynch, "Multigeneration of Transform Coded Images," *Proceedings of SPIE Visual Communications and Image Processing, San Jose, California*, vol. 3024, pp. 1394–1405, February 1997.

[5] O. Werner, "Requantization for Transcoding of MPEG-2 Intraframes," *IEEE Transactions on Image Processing*, vol. 8, pp. 179–191, February 1999.

[6] P. Assunção and M. Ghanbari, "A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 953–967, December 1998.

[7] H. Sorial, W. E. Lynch, and A. Vincent, "Estimating Laplacian Parameters of DCT Coefficients for Requantization in the Transcoding of MPEG-2 Video," *2000 International Conference on Image Processing (ICIP-2000), BC, Canada*, September 2000.

[8] H. Sorial, W. E. Lynch, and A. Vincent, "Selective Requantization for Transcoding of MPEG Compressed Video," *IEEE International Conference on Multimedia and Expo (ICME 2000), NY, USA*, July 2000.

[9] L. Wang and A. Vincent, "Bit Allocation and Constraints for Joint Coding of Multiple Video Programs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 949–959, September 1999.

[10] L. Wang and A. Vincent, "Joint Coding for Multi-Program Transmission," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 425–428, September 1996.

[11] G. Keesman and D. Elias, "Analysis of Joint Bit Rate Control in multi-program Image Coding," *Proceedings of SPIE Visual Communications and Image Processing*, pp. 1906–1917, September 1994.

[12] H. Sorial, W. E. Lynch, and A. Vincent, "Joint Bit-Allocation for MPEG Encoding of Multiple Video Sequences with Minimum Quality-Variation," *IEEE International Symposium on Circuits and Systems (ISCAS'2000), Geneva, Switzerland*, May 2000.

[13] H. Sorial, W. E. Lynch, and A. Vincent, "Joint Transcoding of Multiple MPEG Video Bitstreams," *IEEE International Symposium on Circuits and Systems (ISCAS'99), Orlando, Florida,* vol. 4, pp. 251–254, May 1999.

[14] T. Sikora, "MPEG Digital Video-Coding Standards," *IEEE Signal Processing Magazine*, pp. 82–100, September 1997.

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

[16] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures.* Kluwer Academic Publishers, 1995.

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

[18] ITU-T Recommendation H.261, *Video Codec for Audiovisual Services at p x 64 kbit/s*, March 1993.

[19] ITU-T Recommendation H.263, *Video Coding for Low Bit Rate Communication*, March 1996.

[20] 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 - Video*, Geneva, 1993.

[21] ISO/IEC 13818-2, *Generic Coding of Moving Pictures and Associated Audio Information: Video*, May 1994.

[22] *ISO/IEC JTC1/SC29/WG11 N2725, MPEG-4 Overview - (Seoul Version)*, March 1999.

[23] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 19–31, February 1997.

[24] U. Black, *ATM: Foundation for Broadband Networks.* Prentice Hall Series in Advanced Communication Technology, Prentice Hall, 1995.

[25] ISO/IEC JTC1/SC29/WG11 N1733, *MPEG-7: Context and Objectives, Version 4*, July 1997.

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

[27] L. Torres-Urgell and R. L. Kirlin, "Adaptive Image Compression using Karhunen-Loeve Transform," *Signal Processing*, vol. 21, pp. 303–313, December 1990.

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

[29] N.Ahmed, T.Natrajan, and K. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C-23, pp. 90–93, December 1984.

[30] R. Schafer and T. Sikora, "Digital Video Coding Standards and Their Role in Video Communications," *Proceedings of the IEEE*, vol. 83, pp. 907–923, 1995.

[31] R. J. Clarke, "Relation Between the Karhunen-loeve and Cosine Transforms," *Proceedings of the IEEE*, vol. 128, pp. 359–360, 1981.

[32] F. Dufaux and F. Moscheni, "Motion Estimation Techniques for Digital TV: A Review and A New Contribution," *Proceedings of the IEEE*, vol. 83, pp. 858–876, June 1995.

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

[34] H.Musmann, P. Pirsch, and H. Grallert, "Advances in Picture Coding," *Proceedings of the IEEE*, vol. 73, pp. 523–548, April 1985.

[35] H. Lohscheller, "A Subjectively Adapted Image Communications System," *IEEE Transactions on Communications*, vol. T-COMM 32, pp. 1316–1322, 1984.

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

[37] D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," *Proceedings IRE*, vol. 40(9), pp. 1098–1101, September 1952.

[38] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

[39] ISO/IEC-JTC1/SC29/WG11, MPEG 93/457, *Test Model 5*, April 1993.

[40] G. Keesman, I. Shah, and R. Gunnewiek, "Bit-Rate Control for MPEG Encoders," *Signal Processing: Image Communication*, vol. 6, pp. 545–560, March 1995.

[41] W. Ding and B. Liu, "Rate Control of MPEG Video Coding and Recording by Rate-Quantization Modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 12–20, February 1996.

[42] W. Ding, "Joint Encoder and Channel Rate Control of VBR Video over ATM Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 266–278, February 1997.

[43] J. Black, W. Pennebaker, C. Fogg, and D. LeGall, *MPEG Video Compression Standard.* Digital Multimedia Standards Series, Chapman and Hall, 1996.

[44] K. Rao and J. Hwang, *Techniques and Standards for Image, Video and Audio Coding.* Prentice Hall, 1996.

[45] ISO/IEC JTC1/SC29/WG11 N1642, *MPEG-4 Video Verification Model Version 7.0*, April 1997.

[46] A. Puri and A. Wong, "Spatial Domain Resolution Scalable Video Coding," *Proceedings of SPIE Visual Communications and Image Processing, Boston, MA.*, November 1993.

[47] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, vol. COM-31, pp. 532–540, 1983.

[48] ISO/IEC JTC1/SC29/WG11 Doc. MPEG95/254, *Experimental Results of Coding of Stereo Sequences with Temporal Scalability*, July 1995.

[49] A. Puri, V. Kollarits, and B. Haskell, "Stereoscopic Video Compression Using Temporal Scalability," *Proceedings of SPIE Visual Communications and Image Processing, Taiwan*, May 1995.

[50] A. K. Jain, *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[51] F. Pereira and T. Alpert, "MPEG-4 Video Subjective Test Procedures and Results," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, February 1997.

[52] A. B. Watson, *Digital Images and Human Vision.* MIT Press, 1993.

[53] J. Saghri, P. Cheatham, and A. Habibi, "Image Quality Measure based on a Human Visual System Model," *Journal of Optical Engineering*, vol. 28, pp. 813–818, July 1989.

[54] P. Teo and D. Heeger, "Perceptual Image Distortion," *Proceedings of the 1st IEEE International Conference on Image Processing, Austin, Texas*, vol. 1, pp. 982–986, 1994.

[55] S. Karunasekera and N. Kingsbury, "A Distortion Measure for Blocking Artifacts in Images based on Human Visual Sensitivity," *IEEE Transactions on Image Processing*, vol. 4, pp. 713–724, June 1995.

[56] S. Comes and B. Macq, "Image Quality Criterion based on the Cancellation of the Masked Noise," *ICASSP-95*, vol. 1, pp. 2635–2638, May 1995.

[57] A. Webster, C. Jones, M. Pinson, S. Voran, and S. Wolf, "An Objective Video Quality Assessment System based on Human Perception," *SPIE Human Vision, Visual Processing, and Digital Display*, 1994.

[58] K. Cornog, "Factors in Preserving Video Quality in Post-Production when Cascading Compressed Video Systems," *SMPTE Journal*, pp. 95–102, February 1997.

[59] C. Ricken, "The 4:2:2 Profile of MPEG-2 for Use in a Studio Environment," *SMPTE Journal*, pp. 401–405, July 1996.

[60] *ISO/IEC 13818-2 Amendment 2, Generic Coding of Moving Pictures and Associated Audio*, January 1996.

[61] W. Luo and M. E. Zarki, "Analysis of Error Concealment Schemes for MPEG-2 Video Transmission over ATM Networks," *Proceedings of SPIE Visual Communications and Image Processing, Taipei, Taiwan*, pp. 1358–1368, May 1995.

[62] L. Kieu and N. Ngan, "Cell Loss Concealment Techniques for Layered Video Codec in an ATM Network," *Computers and Graphics: Image Communication*, vol. 18, pp. 11–19, January-February 1994.

[63] P. Salama, N. Shroff, E. Coyle, and E. Delp, "Error Concealment Techniques for Encoded Video Streams," *Proceedings of the IEEE International Conference on Image Processing (ICIP), Washington D.C.*, pp. 9–12, October 1995.

[64] P. Salama, N. Shroff, and E. Delp, "A Fast Suboptimal Approach to Error Concealment in Encoded Video Streams," *Proceedings of the IEEE International Conference on Image Processing (ICIP),Santa Barbara, CA*, October 1997.

[65] Q. Zhu, Y. Wang, and L. Shaw, "Coding and Cell Loss Recovery in DCT Based Packet Video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 248–258, June 1993.

[66] H. Sun and W. Kwok, "Concealment of Damaged Block Transform Coded Images Using Projections onto Convex Sets," *IEEE Transactions on Image Processing*, vol. 4, pp. 470–477, April 1995.

[67] K. Shen, G. Cook, L. Jamieson, and E. Delp, "An Overview of Parallel Processing Approaches to Image Compression," *Proceedings of SPIE Visual Communications and Image Processing, San Jose, CA*, vol. 2186, pp. 197–208, February 1994.

[68] R. Gove, "The MVP: A Highly-Integrated Video Compression Ship," *Proceedings of IEEE Data Compression Conference, Snowbird, Utah*, pp. 28–31, March 1994.

[69] K. Shen and E. Delp, "A Parallel Implementation of an MPEG Encoder," *Proceedings of SPIE Conference on Digital Video Compression: Algorithms and Technologies, San Jose, CA*, vol. 2186, pp. 407–418, February 1995.

[70] S. Akramullah, I. Ahmad, and M. Liou, "A Data-Parallel Approach for Real-Time MPEG-2 Video Encoding," *Journal of Parallel and Distributed Computing*, vol. 30, pp. 129–146, November 1995.

[71] S. M. S and A. Zakhor, "An Optimization Approach for Removing Blocking Effects in Transform Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 74–82, April 1995.

[72] C. Avril and T. Nguyen, "Linear Filter for Reducing Blocking Effects in Orthogonal Transform Image Coding," *Journal of Electronic Imaging*, vol. 1, pp. 183–191, April 1992.

[73] M. Pereira and A. Lippman, "Re-codable Video," *IEEE Proceedings International Conference on Image Processing*, vol. 2, pp. 952–956, November 1994.

[74] Y. Nakajima, H. Hori, and T. Kanoh, "Rate Conversion of MPEG Coded Video by Re-quantization Process," *IEEE Proceedings International Conference on Image Processing*, vol. 3, pp. 408–411, October 1995.

[75] R. J. Safranek, C. R. Kalmanek, and R. Garg, "Methods for Matching Compressed Video to ATM Networks," *IEEE Proceedings International Conference on Image Processing*, vol. 1, pp. 13–16, October 1995.

[76] P. Assunção and M. Ghanbari, "Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates," *IEEE International Conference on Accoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 1998–2001, May 1996.

[77] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG Bitstreams," *Signal Processing: Image Communication*, vol. 8, pp. 481–500, 1996.

[78] J. Youn, M. Sun, and C. Lin, "Motion Vector Refinement for High-Performance Transcoding," *IEEE Transactions on Multimedia*, vol. 1, pp. 30–40, March 1999.

[79] R. Reiniger and J. Gibbs, "Distribution of the Two Dimensional DCT Coefficients for Images," *IEEE Transactions on Communications*, vol. COMM-31, pp. 835–839, June 1983.

[80] H. V. Poor, *An Introduction to Signal Detection and Estimation.* Springer-Verlag, 1988.

[81] A. Guha and D. J. Reininger, "Multichannel Joint Rate Control of VBR MPEG Encoded Video for DBS Applications," *IEEE Transactions on Consumer Electronics*, vol. 40, pp. 616–623, August 1994.

[82] S. C. Liew, , and H. H. Chan, "Lossless Aggregation: A Scheme for Transmitting Multiple Stored VBR Video Streams Over A Shared Communications Channel Without Loss of Image Quality," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1181–1189, August 1997.

[83] L. Wang and A. Vincent, "Bit Allocation for Joint Coding of Multiple Video Programs," *Proceedings of SPIE Visual Communications and Image Processing, San Jose, CA*, pp. 149–158, February 1997.