Joint MPEG-2 Coding for Multi-Program Broadcasting of Pre-Recorded Video

by

Irene Koo

B. A. Sc. (Electrical Engineering), The University of British Columbia, 1995

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

August 1998

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical and Computer Engineering

The University of British Columbia

2356 Main Mall

Vancouver, Canada

V6T 1Z4

Date: Aug 26, '00

Abstract

The tradeoff between picture quality and bandwidth usage is one of the most prominent issues in the world of broadcasting. Since broadcasters are able to simultaneously transmit multiple streams in a channel, they face the challenge of guaranteeing a certain picture quality required by each of the video streams transmitted while maximizing the number of video streams carried in each channel. To address this problem, we developed an MPEG-2 based multi-program video coding system, suitable for digital TV broadcasting, video on demand, and high definition TV over broadcast satellite networks with limited bandwidth. The system can be easily implemented for commercial use in digital broadcasting applications. Compared to present broadcast systems and for the same level of guaranteed picture quality, our system greatly increases the number of video streams transmitted in each channel. As a result, a large number of transponders can be freed to carry real-time broadcasting. By switching from tape storage to video server technology, the need for numerous playback (VTR) systems at the headend is eliminated. In addition, the majority of the complete MPEG-2 encoders are replaced by much less complex MPEG-2 transcoders. The freeing of numerous transponders, the elimination of numerous playback systems, and the replacement of the complete MPEG-2 encoders with MPEG-2 transcoders provide a much more cost-effective solution for the broadcast stations.

Table of Contents

JOINT MPEG-2 CODING FOR MULTI-PROGRAM BROADCASTING OF PRE-RECORDED VIDEO i

ABSTRACT ii

TABLE OF CONTENTS iii

LIST OF FIGURES v

LIST OF TABLES vii

ACKNOWLEDGEMENTS viii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 BACKGROUND 8

2.1 MPEG-2 OVERVIEW 9

2.1.1 MPEG-2 Video Stream Syntax 9

2.1.2 MPEG-2 Video Coding 11

2.2 RELATED WORKS 13

2.2.1 Video Coding of a Single Source 14

2.2.2 Video Coding of Multiple Sources 18

2.2.2.1 Statistical Multiplexing 18

2.2.2.2 Joint Bit-rate Control 20

2.3 SUMMARY 23

CHAPTER 3 TWO-STAGE JOINT BIT-RATE CODING 25

3.1 AN OVERVIEW OF THE TWO-STAGE JOINT BIT-RATE CODING SYSTEM 28

3.2 FIRST STAGE ENCODER 30

3.2.1 Fixed Quantization 31

3.2.2 Video Server: Trade-off between Cost and Picture Quality 37

3.2.3 Complexity Files 46

3.3 JOINT BIT-RATE CONTROLLER 47

3.3.1 Admission Test 47

3.3.2 Processing Procedure for a Group of Pictures 49

3.3.3 Parameters for Optimizing Transcoders' Performance 59

3.3.3.1 First Parameter Set: Picture Complexities 59

3.3.3.2 Second Parameter Set: Target Picture Bit-Rates 60

3.4 TRANSCODERS 60

3.4.1 The Transcoding Processing 62

3.4.1.1 Initialization 62

3.4.1.2 Transcoding Procedure for a Group of Pictures 62

3.4.2 Picture Bit-Rate Distribution 65

3.4.2.1 Transcoding with Given Picture Complexities 65
3.4.2.2 Transcoding with Given Picture Bits Distribution 66
3.5 SUMMARY 73
CHAPTER 4 SIMULATION RESULTS AND DISCUSSIONS 74
4.1 SETUP 75
4.2 SIMULATION RESULTS 75
4.3 TIMING ANALYSIS 88
4.4 SUMMARY 90
CHAPTER 5 SUMMARY AND FUTURE WORK 91
5.1 SUMMARY 91
5.2 FUTURE WORK 93
CHAPTER 6 BIBLIOGRAPHY 94

List of Figures

FIGURE 1.1: TODAY'S BROADCAST SYSTEM. 4

FIGURE 1.2: MULTI-CHANNEL COMPRESSION SYSTEM. 5

FIGURE 2.1: RELATIONSHIP BETWEEN I-, P-, AND B- PICTURES IN A GROUP OF PICTURES. 10

FIGURE 2.2: BLOCK DIAGRAM OF MPEG VIDEO ENCODING. 12

FIGURE 3.1: BLOCK DIAGRAM OF THE TWO-STAGE JOINT BIT-RATE CODING SYSTEM. 28

FIGURE 3.2: BLOCK DIAGRAM OF THE FIRST-STAGE VIDEO ENCODER. 31

FIGURE 3.3: DEFAULT QUANTIZER MATRICES DEFINED IN THE MPEG-2 VIDEO SPECIFICATION. 31

FIGURE 3.4: PICTURE QUALITY (PSNR) OF TEST SEQUENCES ENCODED USING 3 FIXED QUANTIZER SCALES AND THE DEFAULT QUANTIZER MATRICES. 33

FIGURE 3.5: THE MODIFIED INTRA QUANTIZER MATRIX 34

FIGURE 3.6: PICTURE QUALITY (PSNR) COMPARISONS OF TEST SEQUENCES ENCODED WITH THE DEFAULT AND THE MODIFIED INTRA QUANTIZER MATRIX. 36

FIGURE 3.7: PSNR-Q AND R-Q CURVES OF MICROSOFT ROBOT. 39

FIGURE 3.8: PSNR-Q AND R-Q CURVES OF GARDEN 40

FIGURE 3.9: PICTURE QUALITY COMPARISONS AMONG MICROSOFT ROBOT ENCODED AT FIXED QUANTIZER SCALES 7, 8, AND 10. 43

FIGURE 3.10: PICTURE QUALITY COMPARISONS AMONG GARDEN ENCODED AT FIXED QUANTIZER SCALES 7, 8, AND 10. 45

FIGURE 3.11: DATAFLOW DIAGRAM OF THE JOINT BIT-RATE CONTROL PROCESS. 48

FIGURE 3.12: FLOW DIAGRAM OF GOP TARGET ADJUSTMENT ALGORITHM. 52

FIGURE 3.13: GOP BIT ALLOCATION PERFORMED BY THE JOINT BIT-RATE CONTROLLER FOR MICROSOFT ROBOT, BALLET, BUS, AND GARDEN. 57

FIGURE 3.14: GOP BIT ALLOCATION PERFORMED BY THE JOINT BIT-RATE CONTROLLER FOR SEGMENT 1, SEGMENT 2, SEGMENT 3, AND SEGMENT 4 OF THE MAN IN THE IRON MASK TRAILER 59

FIGURE 3.15: BLOCK DIAGRAM OF THE TRANSCODING PROCESS. 61

FIGURE 3.16: DATAFLOW DIAGRAM OF THE TRANSCODING PROCESS. 63

FIGURE 3.17: THE PICTURE QUALITY OF THE MAN IN THE IRON MASK TRAILER SEGMENTS TRANSCODED WITH GOP TARGETS ONLY AND THE PICTURE QUALITY OF THE SAME SEGMENTS TRANSCODED WITH GOP TARGETS AND PICTURE COMPLEXITIES. 70

FIGURE 3.18: THE PICTURE QUALITY OF MAN IN THE IRON MASK TRAILER SEGMENTS TRANSCODED WITH GOP TARGETS ONLY AND THE PICTURE QUALITY OF THE SAME SEGMENTS TRANSCODED WITH GOP AND PICTURE TARGETS. 72

FIGURE 4.1: GOP COMPLEXITIES AND BIT-RATES OF THE MAN IN THE IRON MASK VIDEO SEQUENCES. 81

FIGURE 4.2: GOP COMPLEXITIES AND BIT-RATES OF THE SECOND SET OF VIDEO SEQUENCES. 84

List of Tables

TABLE 3.1: PSNR VALUES OF TEST SEQUENCES ENCODED WITH THE DEFAULT AND PROPOSED INTRA QUANTIZER MATRICES. 35

TABLE 3.2: COMPARISON OF PSNR VALUES FOR VIDEO STREAMS ENCODED WITH JOINT BIT-RATE CODING USING GOP TARGETS ONLY, WITH JOINT BIT-RATE CODING USING GOP TARGETS AND PICTURE COMPLEXITIES, AND WITH JOINT BIT-RATE CODING USING GOP AND PICTURE TARGETS. 68

TABLE 3.3: BIT-RATES OF THE FOUR MAN IN THE IRON MASK VIDEO SEGMENTS ASSIGNED BY THE JOINT BIT-RATE CONTROLLER. 68

TABLE 4.1(A): BIT-RATES OF THE MAN IN THE IRON MASK VIDEO SEQUENCES ENCODED USING THE JOINT BIT-RATE CODING SYSTEM. 77

TABLE 4.1(B): AVERAGE GOP COMPLEXITY OF THE MAN IN THE IRON MASK VIDEO SEQUENCES. 77

TABLE 4.2(A): BIT-RATES OF THE VIDEO SEQUENCES FROM THE SECOND TEST SET ENCODED USING THE JOINT BIT-RATE CODING SYSTEM. 78

TABLE 4.2(B): AVERAGE GOP COMPLEXITY OF THE VIDEO SEQUENCES FROM THE SECOND TEST SET. 78

TABLE 4.3: PSNR STANDARD DEVIATIONS FOR MAN IN THE IRON MASK VIDEO SEQUENCES ENCODED USING OUR TWO-STAGE JOINT BIT-RATE CODING SYSTEM AND ENCODED INDEPENDENTLY USING THE TM5 METHOD. 86

TABLE 4.4: PSNR STANDARD DEVIATIONS FOR THE SECOND SET OF VIDEO SEQUENCES ENCODED USING OUR TWO-STAGE JOINT BIT-RATE CODING SYSTEM AND ENCODED INDEPENDENTLY USING THE TM5 METHOD. 87

TABLE 4.5: CPU TIME USED IN TRANSCODING AND ENCODING THE MAN IN THE IRON MASK VIDEO SEQUENCES. 89

TABLE 4.6: CPU TIME USED IN TRANSCODING AND ENCODING THE SECOND SET OF VIDEO SEQUENCES. 89

Acknowledgements

I owe my greatest debt to my supervisors, Professor Rabab Ward and Professor Panos Nasiopoulos, for guiding me from the initial development of the project to the writing of this thesis and for giving me invaluable advice and feedback. Without their support, this thesis could not have been written.

I would like to thank my parents for their patience and encouragement and for their understanding of my not being at home all the time. Many thanks to my sister, Jenny, for taking care of mom and dad when I spent much of my time at school and for introducing me to a wonderful group of friends.

I wish to express my greatest gratitude to Amy, Grace, Precy, and Rebecca. They have given me friendships since the days of my undergraduate studies. I cherish every moment we spent together talking, laughing, as well as comforting each other. Hopefully, we will have many more gatherings like we used to have.

I would also like to take this opportunity to thank a group of friends who made the past year an enjoyable experience. Ice-skating, birthday celebrating, playing board games, seeing movies, barbecuing, having fun at Playland, watching Symphony of Fire, and of course, golfing (How can we ever forget that!) have created lasting memories.

I also want to express special thanks to Miranda and all the friends in the Image Processing Lab as well as those in the Communications Lab. They have been very helpful when I presented various ideas and problems to them.

And to all other people who cares about my well being, "Thank you!"

Chapter 1 Introduction

The introduction of the MPEG-1 standard in 1991 and MPEG-2 in 1993 has led to an increasing number of market opportunities for new communication services that provide their customers with more flexibility, capability, and convenience than the existing systems. One of the newer applications resulting from the standardization of MPEG is Digital Satellite Broadcast (DSB). Analog satellite broadcasting has been around since 1962, the year when the first satellite TV transmission via Telstar I, an eight-minute experimental broadcast from France to the U.S., occurred. In 1976, Taylor Howard of San Andreas, California, became the first individual to receive C-band satellite TV signals on a homebuilt system [24]. However due to the high cost of hardware, it was not till 1980 that the first home satellite system priced below US $10,000 became available. By 1989, the price of a home satellite system dropped to around US$2000.00, and the industry was able to ship only 383,000 units. However, after the demonstration of digital video compression by General Instrument and SkyPix in 1991 and the introduction of new U.S. legislation guaranteeing access to satellite-delivered cable programming services by alternative multi-channel video providers such as Digital Broadcast Satellite (DBS) operators, the number of shipped home satellite systems sky-rocketed to 4 million units by 1993 [24]. Currently, at least three major Direct-to-Home Satellite Broadcasting companies are providing DSB services in the United States and together offering more than 540 channels of programming [25]. Today, there are over 7.3 million DSB subscribers in the United States [28].

Another application that makes exclusive use of the MPEG-2 standard is Digital Television Broadcast (DTV). Discussions about providing digital television broadcasting services in the U.S. has begun in 1987 when 58 broadcasters requested the Federal Communications Committee (FCC) to establish a broadcast standard for high-resolution TV services [26]. However, due to technical implications and political issues involved, it was not till April 3, 1997, that the FCC was finally able to give its orders for the launch of digital TV. "The FCC requires the affiliates of the top four networks in the top ten markets to be on-the-air with a digital signal by May 1, 1999. Affiliates of the top four networks in markets 11-30 must be on-the-air by November 1, 1999" [29]. When applying a modulation scheme such as Vestigial Side Band (VSB) or Quadrature Amplitude Modulation (QAM) on the 6 MHz analog channel used today, the channel can sustain 19.3 Mbps of data. This bandwidth can accommodate a High Definition TV (HDTV) video program or five Standard Definition TV (SDTV) programs. A HDTV signal has about four times the resolution of today's NTSC signal while a STDV signal has a resolution similar to today's NTSC signal. Since both DSB and DTV have the ability to support multiple programs in a single channel, broadcasters are given the flexibility as well as the challenges in coding and transmitting multiple sources down a fixed bandwidth channel.

A typical digital satellite broadcast system consists of three parts: uplink earth stations (headend), distribution satellites, and downlink earth stations. Each transponder in a satellite has the bandwidth to support multiple transmission channels, and each transmission channel can carry multiple video streams. The uplink earth station is where the encoding and multiplexing of video streams take place. The video coding system located in an uplink earth station comprises a set of video recording and playback systems, a video cassette archive, a cassette library management system, a scheduling system, and a set of multi-channel compression systems (see Figure 1.1). A multi-channel compression system usually consists of a set of encoders, a channel controller, a multiplexer, and a supervisory system (see Figure 1.2). Based on the program schedules generated by the scheduling system, the video programs to be broadcast soon are retrieved from the archive and placed in the set of playback systems. Real-time video programs are also played back immediately. The signal out of each playback system is fed into a compression system. The necessary service parameters of a video program such as bit-rate are set according to the contracted quality of service (QoS), the video program itself, and the coexisting video programs in the channel. The assigned bit-rate does vary when the coding environment (e.g., the number of video programs coexisted in the channel), the characteristics of the video program or the characteristics of other coexist video programs change. However, the assigned bit-rate usually stays constant for a long period of time before it is changed. After parameter specifications, the video programs are then encoded. The channel controller controls the multiplexing of the video streams.

Several drawbacks have been recognized in the current broadcast systems:

1. Inefficient use of bandwidth resulting from constant bit-rate (CBR) coding. Much work has already proven that CBR coding is inherently inefficient in terms of bandwidth usage.

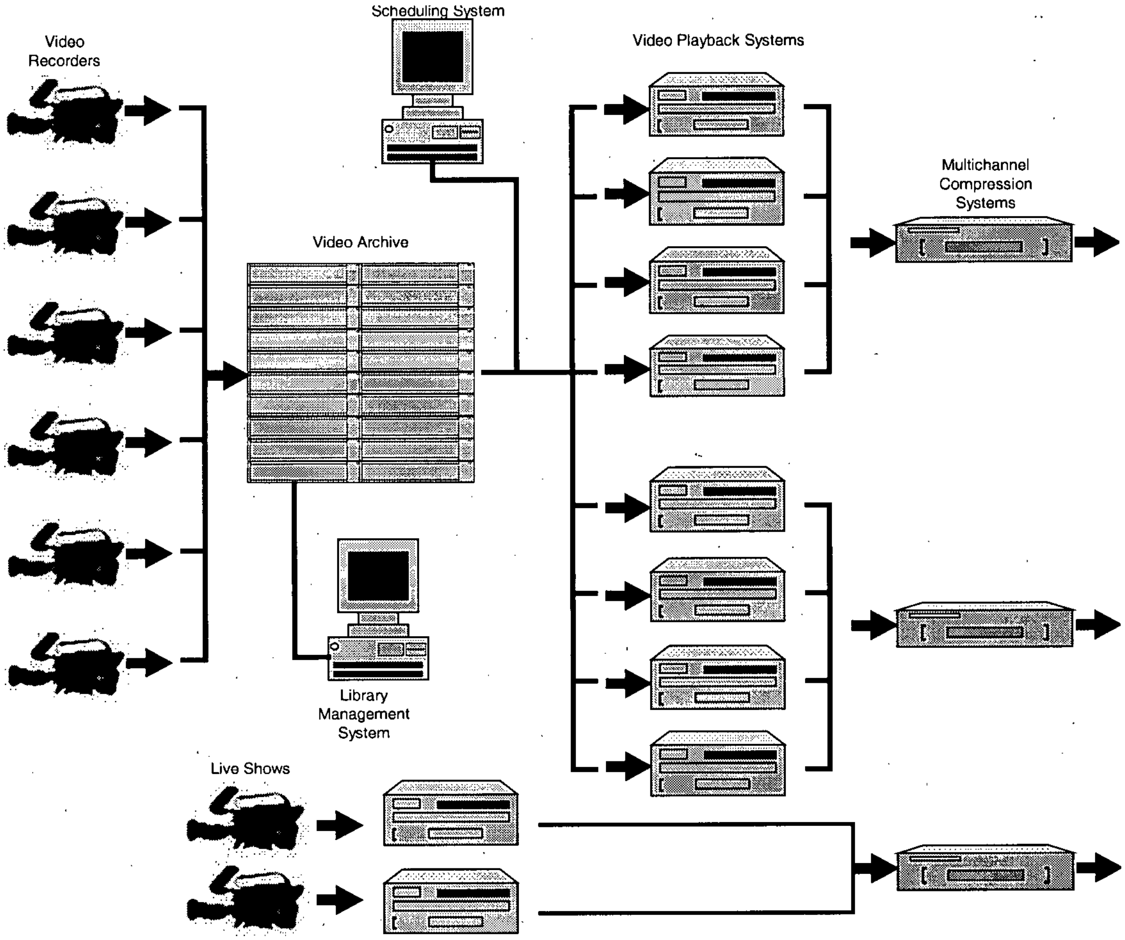


Figure 1.1: Today's broadcast system.

2. Large picture quality fluctuation resulting from CBR coding. In addition to poor bandwidth usage, CBR coding also generates large fluctuations in picture quality.
3. Problems with the use of videocassette tapes for archiving. One video playback system is needed for each pre-recorded video program to be transmitted. Although a videocassette tape is a storage medium with huge storage capacity, it is also a medium with no random access supports. Video tapes are subject to physical deterioration when they are used frequently. The sharing of videotape cannot be easily done when several people try to gain access to the same tape simultaneously.

Multiple copies of the tape are usually needed to accommodate the demands. Most importantly, videocassette tapes require large physical storage space.

4. The need for numerous encoders and the high cost of the broadcast systems. Since video programs are encoded immediately before transmission, one encoder is needed for each of the video programs to be transmitted. The high cost of the encoders increases the cost of a broadcast system substantially.

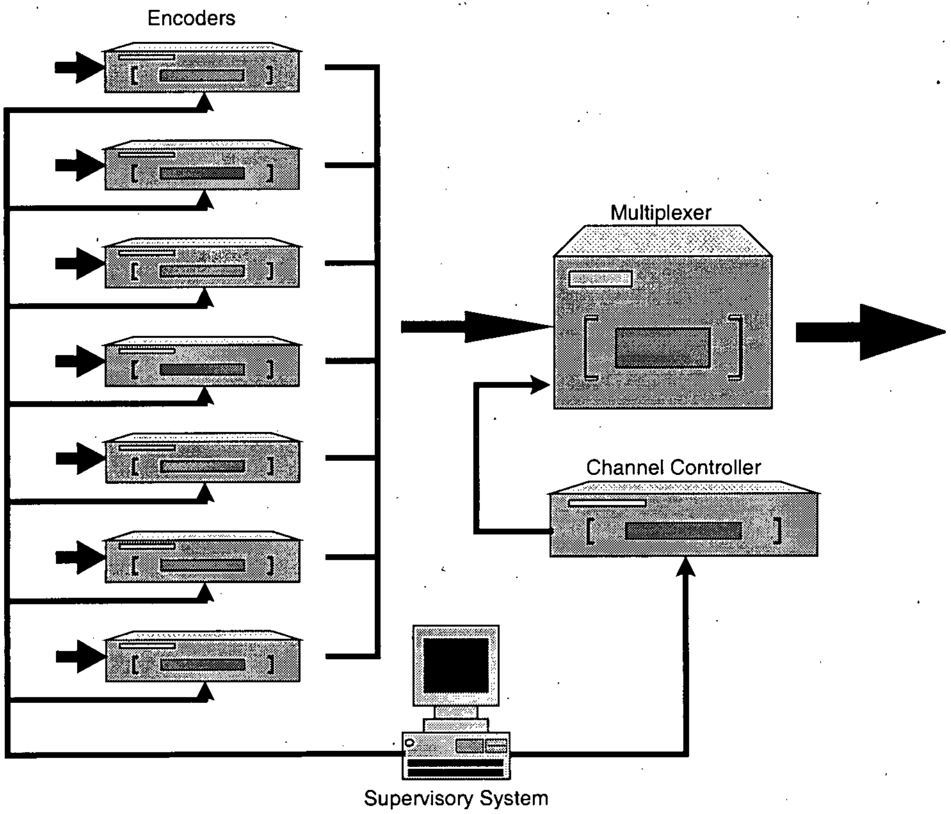


Figure 1.2: Multi-channel compression system.

The goal of this thesis is to develop a simple yet effective multi-channel video coding system. The system should address the problems encountered by today's broadcast systems. More specifically, the coding system

1. should allocate bandwidth to each video program efficiently,

2. should offer a consistent picture quality,
3. should have a lower cost than the cost of today's systems, and
4. should not use cassette tape for archiving.

In order to meet the above requirements, we propose for our system

1. the use of a bit allocation scheme that assigns to each video stream only the necessary number of bits and minimizes the fluctuations in picture quality,
2. the use of a set of transcoders, which are structurally less complex than an encoder, for encoding the video programs immediately before transmission, and
3. the use of a video server for storing video programs.

We develop a two-stage joint bit-rate coding system that has all the characteristics just described. The system is product-oriented and can be easily implemented for commercial use. The system has three components: a video encoder and a video server, a joint bit-rate controller, and a set of transcoders. The system performs multi-channel video encoding in two stages. During the first stage, the system performs high bit-rate – high quality video encoding on the video sources. During the second stage, it converts the high bit-rate video streams to lower bit-rate video streams. When the lower bit-rate video streams are multiplexed, the resulting stream satisfies the network bandwidth requirement. Traditionally, the complexity of a video stream is determined on a picture basis, and the complexity of a picture is estimated from the statistics of some other encoded picture. Such estimation is clearly not accurate in most cases. For example, the complexities of the pictures immediately before and immediately after a scene change are obviously uncorrelated. To overcome the picture complexity estimation problem, our system measures and records the picture complexities from actual data while it is performing the encoding during the first stage. To simplify the bandwidth allocation decisions in the second stage, we also define a group of pictures (GOP) complexity measure that indicates the complexity of a GOP. By utilizing the recorded GOP complexities and picture complexities, our joint bit-rate control method distributes the necessary bandwidth to each video source. Finally, the set of transcoders converts the encoded high bit-rate video streams to ones that have their bit-rates determined using the joint bit-rate control method.

This system guarantees the efficient use of the available bandwidth. For the same level of contracted QoS, it significantly increases the number of transmitted video streams per channel. Compared to existing applications, this implementation offers a very cost-effective solution by greatly reducing the number of playback systems as well as “complete” MPEG-2 encoders needed at the broadcasting headend and by “freeing” transponders for real-time broadcasting.

The organization of the thesis is as follows: in Section 2 we give an overview of the MPEG video encoding process and discuss some of the related works. In Section 3, we describe our two-stage joint bit-rate coding solution.. In Section 4, we present the simulation results of our system. Finally, in Section 5 we provide a summary and possible future work.

Chapter 2 Background

MPEG is an audio-visual communication standard that is found in many applications. The first version of MPEG, MPEG-1 [23], was designed for digital medium storage of audio-visual signals at 1.5 Mbps. The video coding of MPEG-1 was targeted for sources with SIF resolution (352x240 at 30 non-interlaced frames/s or 352x288 at 25 non-interlaced frames/s) at bit-rates of about 1.2 Mbps while the audio coding was targeted at bit-rates around 250 kbps [3]. MPEG-2 is an extension to MPEG-1. It is more generic and has more features and capabilities than MPEG-1. MPEG-2 is designed for both digital medium storage and transmission and intended for interlaced CCIR601 source at about 4 Mbps. However, MPEG-2 supports bit-rates at high as 429 Gbps [3]. MPEG-2 is found in numerous applications. Digital Versatile Disc (DVD), Digital Video Broadcasting (DVB) and Digital TV (DTV) are three of the more prominent examples. The focus of this chapter is to give the reader an overview of MPEG-2 video coding and to describe some of the works that have been done on this subject.

Chapter 2 is organized as follows: in Section 2.1, we outline the structure of an MPEG-2 coded bit stream and briefly review the fundamental concepts underlying the MPEG-2 video coding standard. In Section 2.2, we present some of the previous works done in the MPEG-2 video coding of a single source as well as the simultaneous coding of multiple sources.

2.1 MPEG-2 Overview

The MPEG-2 audio-visual coding standard currently consists of 9 parts, covering different aspects such as systems, video, audio, compliance testing, and real-time interface for system decoders. Part two of the standard, which we refer to as the MPEG-2 video specification [2], specifies the syntax and the decoding semantic for an MPEG-2 compliant video stream. The specification does not specify the encoding process for MPEG-2 video streams. It is up to the system designers to design systems that produce MPEG-2 compliant video streams.

2.1.1 MPEG-2 Video Stream Syntax

The MPEG-2 video specification puts the information of a video stream into a hierarchical structure that consists of six layers: sequence layer, group of pictures (GOP) layer, picture layer, slice layer, macroblock layer, and block layer. All but the block layer have their own headers, which store information pertaining to their respective layers. For example, the sequence header includes information such as bit-rate and optional quantizer matrices, which are relevant to the entire video stream. The GOP header contains information such as a time code for supporting random access, fast search, and editing [3]. The picture header comprises information such as picture coding type, picture structure, and scan format, which are specific to that particular picture. Three picture coding types are defined in MPEG to exploit spatial redundancy and temporal redundancy that are inherent in any video sequences. The three picture coding types are Intra-coded pictures (I- pictures), Predictive pictures (P- pictures), and Bi-directionally predictive pictures (B- pictures).. I- pictures are coded independently. P- pictures are coded with reference to the most recently coded I- or P- picture. B- pictures are coded with reference to two pictures. One is the most recent I- or P- picture. The other is the first I- or P- picture after the B- picture. These different picture coding types usually appear in a repeating pattern such as IBBPBBPBB in the video sequence. Such a repeating sequence of images makes up a GOP. Figure 2.1 gives an example of the GOP structure and points out the pictures from which the P- pictures and B- pictures are predicted. A slice is a series of consecutive macroblocks that are located on the same row of a picture. It is defined to aid resynchronization in case of transmission errors. The slice layer contains information such as quantizer scale applicable to all the macroblocks in the slice. A macroblock is the smallest codable unit. It is made up of four 8x8 luminance blocks and at least one 8x8 chrominance block. In the macroblock layer, information such as the macroblock type, the quantizer scale, the motion type, the motion vectors, and the macroblock pattern of a macroblock is found. A block consists of 8x8 pixel values of an image. The pixel values in a block are quantized and discrete cosine transformed (DCT).

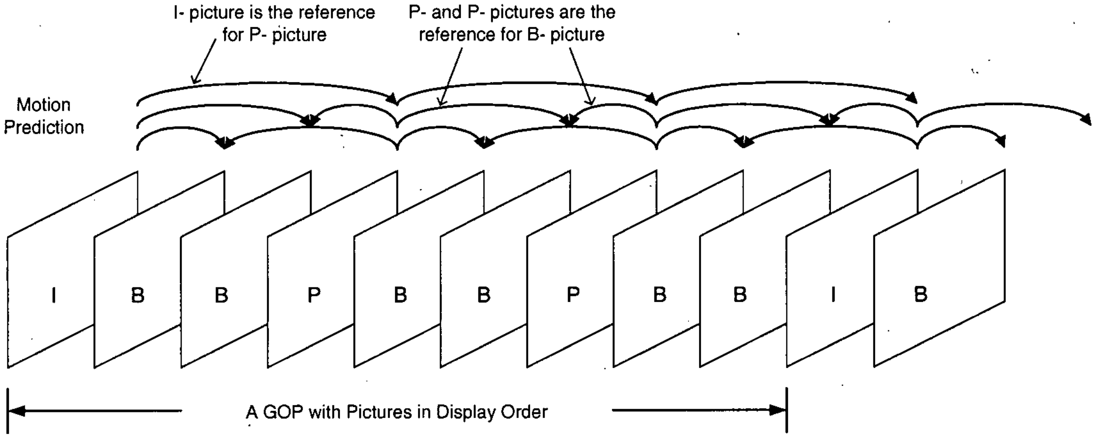

I- picture is the reference for P- picture
P- and P- pictures are the reference for B- picture
Motion Prediction
I
B
B
P
B
B
P
B
B
I
B
A GOP with Pictures in Display Order

Figure 2.1: Relationship between I-, P-, and B- pictures in a Group of Pictures.

2.1.2 MPEG-2 Video Coding

The purpose of video coding is to compress a video sequence such that the resulting video stream has the desired bit-rate. Both MPEG-1 and MPEG-2 use a block-based compression technique that involves both motion compensation prediction and discrete cosine transformations. Figure 2.2 shows the block diagram of a typical MPEG video encoding process. First, the sequence layer and GOP layer information are determined from a user-supplied parameter list and encoded into the output video stream. Also based on the user-supplied parameter list, the picture coding type of each picture is decided upon and the picture layer information is gathered. Each image is then divided into 16x16 pixel macroblocks. Slice layer information and macroblock layer information are determined and encoded. Motion compensation predictions are preformed on the macroblocks if the picture is a P- or B- picture. Each macroblock is then further divided into 8x8 blocks. DCT is applied to each block, and the resulting DCT coefficients are subsequently quantized using both a quantizer scale and an 8x8 quantizer matrix. Finally, the quantized DCT coefficients are variable length encoded, and the resulting video stream is outputted.

The amount of compression resulting from coding depends on the bit-rate desired. In MPEG video coding, compression is achieved by quantizing the blocks of DCT coefficients. Except for the DC term of an intra-coded block, the quantization step-size used in quantizing a DCT coefficient is determined by both the quantizer scale and the corresponding weighting factor in the quantizer matrix. The quantization step-size of the DC term of an intra-coded block depends on the coding parameter *intra_dc_precision*, which is specified by the user. Since the weighting factors of a quantizer matrix are seldom changed during the coding process, bit-rate control is accomplished through changing the quantizer scale value. Two levels of bit-rate control, ***global*** and ***local,*** are performed in MPEG video coding. Global bit-rate control assigns a target number of bits to each picture within a GOP. Based on the target number of bits for the picture and other information, a quantizer scale is determined. In local bit-rate control, this quantizer scale is refined for each macroblock in the picture so that the resulting number of bits used in coding the picture closely matches the target.

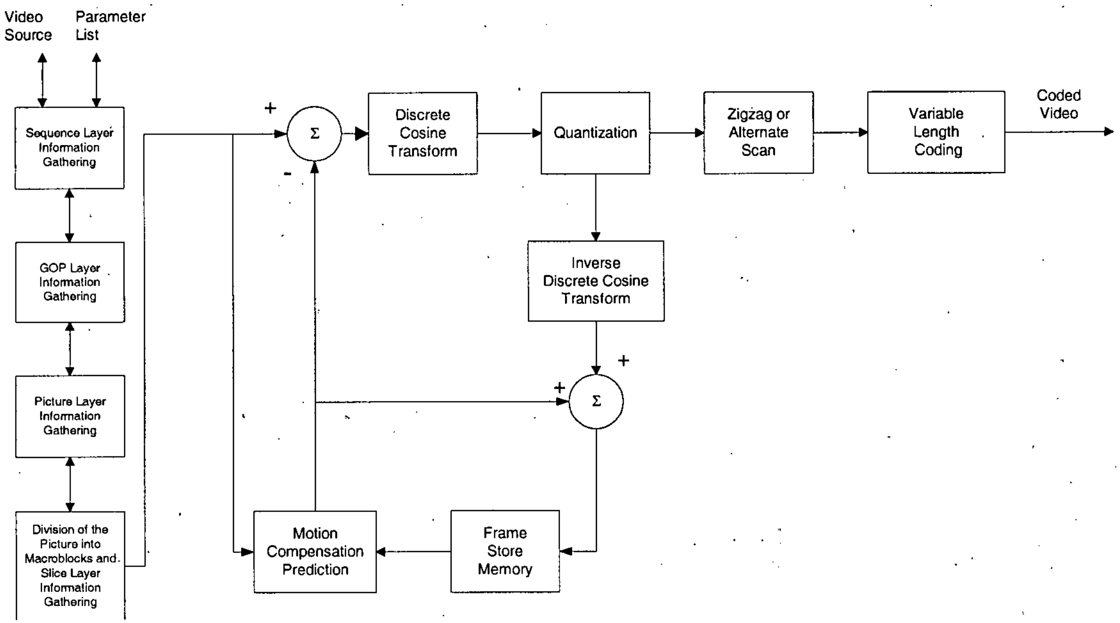

Video Source
Parameter List
Sequence Layer Information Gathering
GOP Layer Information Gathering
Picture Layer Information Gathering
Division of the Picture into Macroblocks and Slice Layer Information Gathering
+
-
Σ
Discrete Cosine Transform
Quantization
Zigzag or Alternate Scan
Variable Length Coding
Coded Video
Inverse Discrete Cosine Transform
+
+
Σ
Motion Compensation Prediction
Frame Store Memory

Figure 2.2: Block diagram of MPEG video encoding.

Two classes of video streams, CBR and VBR, can be generated by MPEG video coding. For a CBR video stream, the bit-rate is regulated by assigning the same number of bits to each GOP, regardless of the GOP's complexity measure, i.e., activity level. If the video stream is given a high bit-rate, some of the assigned bits are wasted when the pictures in the GOP are relatively less complex. On the other hand, if the video stream is given a low bit-rate, the pictures that are more complex suffer from poor picture quality because they have insufficient bits. VBR coding solves the inefficiency in bandwidth usage as well as the inconsistency in picture quality by assigning to individual picture or individual GOP the number of bits it requires to achieve acceptable quality. That is, high-complexity pictures are usually allocated more bits than low-complexity pictures. For the same level of QoS, the resulting VBR video stream has lower average bit-rate requirement and more consistent picture quality than its CBR counterpart.

With a higher channel bandwidth, several video streams can be coded and multiplexed together for transmission. Each video stream uses up only a portion of the available channel bandwidth. In this situation, there usually exists an external mechanism that regulates the bit assignment for each video stream. The bit allocation is usually based on the characteristics of the video sources as well as the network conditions.

2.2 Related Works

Bit allocation is the strategy used by the existing MPEG-2 encoding algorithms for providing consistent picture quality. It is the process of determining a desired number of bits for a GOP and/or for a picture within a GOP. The desired number of bits for a GOP and for a picture is called a ***GOP target*** and a ***Picture target*** respectively [21]. The aim of bit allocation is to achieve picture quality consistency across all the pictures in a GOP. Such consistency can be obtained if each GOP target and each picture target reflect the activity and complexity level of the corresponding GOP and picture. In CBR coding, the number of bits per GOP is fixed. The existing algorithms try to distribute this fixed number of bits to the pictures within the GOP in such a way that each picture target reflects the activity level of the picture and the resulting picture quality is consistent. Unfortunately, an optimum distribution cannot be easily found since the bit-rate of a CBR video stream is tightly controlled. On the other hand, the number of bits required per GOP is not fixed in variable bit-rate coding. The existing algorithms can easily determine an appropriate picture target for each picture with the absence of a bit-rate constraint. Two steps are taken in the bit distribution / bit allocation process. The first step is the determination of the complexity level of a picture. The other is the mapping of this complexity level to a picture target.

2.2.1 Video Coding of a Single Source

The Test Model 5 [4] originated from the MPEG committee defines picture complexity as the product of the average picture quantization factor and the number of bits used to encode the picture. Since bit allocation takes place before the actual encoding of the picture, both the average picture quantization factor and the number of bits used to encode the picture are estimated from the most recently encoded picture of the same picture coding type. The model also defines picture target to be proportional to both the number of bits available for a GOP and the ratio of the picture complexity to a weighted sum of all picture complexities of the GOP.

In [5], Viscito and Gonzales define a *coding difficulty factor* for determining picture targets. The coding difficulty factor of a picture is the sum of the “mean absolute differences from DC” of the intra-coded macroblocks in the picture and the mean absolute prediction differences of the inter-coded macroblocks in the picture. The mean absolute differences from DC for an intra-coded macroblock is given by

$$\begin{aligned}\Delta(r,c) &= \tfrac{1}{4}\sum_{k=0}^{3}\Delta_k(r,c) \\ \Delta_k(r,c) &= \tfrac{1}{64}\sum\sum\left|y_k(i,j)-dc_k\right|\end{aligned} \qquad (2.1)$$

where, r and c are the horizontal and vertical offset counting from the top left corner of a macroblock; k is the number of luminance blocks in a macroblock; Δ_k is the absolute differences from DC of an intra- coded block; $y_k(i,j)$ is the luminance value of the intra-coded block, and dc_k is the DC value of the intra-coded block. The mean absolute prediction differences for an inter-coded macroblock is defined as

$$\begin{aligned}mad(r,c) &= \tfrac{1}{4}\sum_{k=0}^{3} mad_k(r,c) \\ mad_k(r,c) &= \tfrac{1}{64}\sum_{i=0}^{7}\sum_{j=0}^{7}\left|e_k(i,j)\right|\end{aligned} \qquad (2.2)$$

where, $e_k(i,j)$ is the prediction error and $mad_k(i,j)$ is the absolute prediction differences. Finally, the coding difficulty factor of a picture is determined as

$$D = \sum_{macroblock\in\{intra-coded\}} \Delta(r,c) + \sum_{macroblock\in\{inter-coded\}} mad\ \ (r,c) \qquad (2.3)$$

The picture targets for I-, P-, and B- pictures are determined by simultaneously satisfying the following three equations:

$$\begin{aligned}C_{GOP} &= C_I + n_p C_p + n_b C_b \\ C_p &= \frac{D_P - E_d}{D_I} C_I \\ C_b &= w_B \frac{D_P - E'_d}{D_I} C_I\end{aligned} \qquad (2.4)$$

where, D_I, D_P, and D_B are the difficulty coding factors of I-, P-, and B- pictures respectively; n_p and n_b are the number of P- and B- pictures respectively; C_{GOP} is the given GOP target; C_I, C_P, and C_B are the picture targets for I-, P-, and B- pictures; E_d and E'_d are the average mean absolute errors of the past and future decoded pictures to which the P- and B- pictures are referenced; w_B is a weighting factor. Since D_p and D_B are unknown, when the picture targets are being computed, they are estimated from previously encoded P- and B- pictures.

In [6], the picture complexity is defined as the average variance of all 8x8 luminance blocks in a picture. To determine picture targets, the algorithm first separates the video sequence into segments of different scenes and classifies the scenes according to the picture complexity of I-, P-, and B- pictures in the scene. Picture targets are then obtained using pre-computed experimental I-, P-, and B- picture bit counts for each class.

Instead of using a picture complexity measure, [7] uses a macroblock activity measure for picture bit allocation. The macroblock activity measure is defined as the average of the quality bit-count ratios over the quantizer scale index ranges [1, 2,..., 31]. For each quantizer scale, the quality bit-count ratio is determined from the bit-count of the original macroblock and the bit count of the encoded macroblock using this quantizer scale value. A high ratio indicates that the macroblock is easy to encode and thus, requires fewer bits.

[8] uses a feedback re-encoding method to determine picture complexities and picture targets. For each picture, two encodings are performed. During the first encoding, the picture complexity of a picture is estimated from previous I-, P-, or B- pictures in the

same way as in TM5 [4], and a picture target is determined from the estimated picture complexity. The picture is then encoded. The corresponding average quantization factor Q and the number of bits R used in encoding the picture are obtained. The picture complexity is then updated, and the picture target is re-determined using the updated picture complexity.

Another re-encoding method is found in [9]. The algorithm defines macroblock complexity as the number of bits needed to encode a macroblock using a quantizer scale q. The picture complexity of a picture is defined as the sum of the macroblock complexities from all the macroblocks in the picture. The algorithm first encodes a picture using a single quantizer scale value. Using the resulting bitcount from each macroblock, the picture complexity is determined. The algorithm then distributes a given GOP target to the I-, and P-, and B- pictures in the GOP using two bit allocation ratios. The bit allocation ratios are derived from the picture complexities of the most recent I-, P-, and B- picture.

All above bit allocation techniques address the picture quality consistency issue related to the coding of a single video source. The techniques described in [4], [5], and [6] use statistics from previously encoded pictures or from previously encoded training sequences to make assumptions about the current pictures. Past statistics usually are not good representatives of the activity and complexity levels of current pictures because scene changes occur rather quickly within a video sequence. Both [7] and [8] use re-encoding to solve the problem from using past statistics. By encoding a picture or a sequence more than once, more accurate complexity measures are obtained. However, re-encoding requires extra computations. It also introduces long delay before the transmission of the video sequence.

2.2.2 Video Coding of Multiple Sources

Two classes of techniques, statistical multiplexing and joint bit-rate control, can be used in handling bit allocation for multiple program sources. The aim of these techniques is to assign an appropriate bit-rate to provide consistent picture quality for each video source in the multi-program environment. Generally, the goals are accomplished by encoding the video streams using VBR. Although both techniques support VBR compression in a constant bit-rate medium and make use of this knowledge in performing bit allocation, their bit allocation strategies are very different from each other.

2.2.2.1 Statistical Multiplexing

Statistical multiplexing is usually associated with packet switching or cell switching networks such as an ATM network. It finds applications in Direct Satellite Broadcast (DSB) more often than in terrestrial broadcasting since satellite transmission employs the ATM network protocols. In statistical multiplexing, VBR data from each source are split into fixed size segments or "cells", and the cells are placed in a buffer. Immediately before the transmission, the cells from different sources are extracted from the buffer, randomly interleaved, and multiplexed. Three factors are found to have the greatest impact on picture quality when video streams are transmitted via an ATM network [10, 11]. They are the number of lost cells, the number of pixels in an impaired region and its shape, and the burstiness of the loss. In statistical multiplexing, the aim of bandwidth allocation is to minimize the probability of cell losses.

In an ATM network, the probability of cell loss can be minimized if the network is informed about the behaviour that can be anticipated from each individual source. Therefore, video traffic characteristics are first modeled before the actual bandwidth allocation. After a traffic model is chosen, features or model parameters are extracted from each video source. Based on the values of these parameters, the required bandwidth is estimated. For example, [12] uses a Markov chain to model video traffic. The mean of cells generated per frame, μ, and the standard deviation of cells generated per frame, σ, are the two parameters used in bandwidth estimations. A statistical model, found in [13], uses the average bit-rate, the bit-rate variance, and the peak bit-rate of a video source as parameters to characterize the video source. Using simulation results, [14] shows that a VBR bit-rate video stream can be characterized using statistical measures such as the marginal distributions and the peak-to-average ratio of the bit-rates. A parametric model proposed in [15], uses nine fundamental indexes, which are the average intensity level of each picture, the variance of the intensity levels in each picture, the entropy of the pixel values in each picture, the vertical entropy of the pixel values in each picture, the horizontal entropy of the pixel values in each picture, the pixel value difference between consecutive pictures, the motion index of each picture, the temporal entropy of each picture, and the temporal vertical entropy of each picture, to represent a VBR video source.

Statistical multiplexing is based on the "law of large numbers" [16]. It is very effective when the number of video sources to be multiplexed is large. Although statistical multiplexing has its merits, it also has several pitfalls. Because statistical multiplexing is subject to packet loss, the entire channel cannot be used to its full capacity [17]. If a source provides too much data, causing the buffer to overflow, packets will be lost. Similarly, if data is being queued in the buffer for too long, causing it to arrive late at the decoder, the data will also be considered lost. Thus, the performance of statistical multiplexing depends greatly on the statistical model used. Any deviation of the actual data from the model would create catastrophic effects in performance.

2.2.2.2 Joint Bit-rate Control

Joint bit-rate control is a multi-program rate control technique that can be used in various types of applications such as terrestrial broadcasting, satellite broadcasting, cable transmission, or even ATM transmission. This technique is not associated with any particular type of networks as in the case of statistical multiplexing. The idea of joint bit-rate control is to allow the bit-rate of each individual video program to vary according to some video characteristic such as the picture complexity, while the sum of all bit-rates remains constant. In this technique, each video stream gets allocated a portion of the channel bandwidth at every instance. The aim of joint bit-rate control is to perform bit allocation in such a way that the picture quality of each video stream remains relatively consistent. Since video streams under joint bit-rate control are not subject to packet loss, the full channel capacity can be used.

Almost all of the existing joint bit-rate control techniques distribute the channel bandwidth according to some relative parametrical measures of the video streams. For example, the technique in [18] determines a picture target for each video stream by defining picture complexity to be the same as in TM5 [4] and by using the same quantization scale Q for all video programs at each picture. The picture target of each video stream depends upon the ratio of the stream's picture complexity to the sum of the picture complexities of all video streams. i.e.,

$$R_i(k) = \frac{X_i(k)}{\sum_{n=1}^{G} X_n(k)} R_{t\arg et} \qquad (2.5)$$

where, R$_i$(k) is the picture target for video stream i at picture k; X$_i$(k) is the picture complexity for picture k of video stream i; R$_{target}$ is the sum of the available number of bits to each picture k of all video streams, and G is the number of video streams to be multiplexed.

A similar technique to [18] is found in [17]. The latter models the picture quality VQ of a picture to be exponentially proportional to the ratio of the picture target to the picture complexity. i.e.,

$$VQ = 10(1 - e^{-R/X}) \text{ or } R = -X \times \ln(1 - \tfrac{VQ}{10}) \qquad (2.6)$$

where, R is the picture target, and X is the picture complexity. By setting the picture quality for all video streams to be equal, the average picture target $\overline{R}$ satisfies the equation $\frac{\overline{R}}{\overline{X}} = -\ln(1 - \frac{VQ}{10})$. The individual picture target R$_i$ is proportional to the ratio of the picture complexity to the average picture complexity. i.e.,

$$R_i = \frac{X_i}{\overline{X}} \overline{R} \qquad (2.7)$$

Instead of comparing picture complexities, the technique in [19] allocates to the picture of each video stream a bandwidth that is proportional to the video stream's traffic. The traffic of a picture is characterized by two parameters– the reference bandwidth

BW_{PCTi} and the estimated bandwidth requirement BW_{ESTi}. The reference bandwidth of a picture is determined by the total bandwidth available to the pictures of all video streams, the picture coding type, the GOP structure of each video stream, and the current state of the total virtual buffer. The estimated picture bandwidth is based on the picture coding type. For I- pictures, it is determined by encoding its AC coefficients with a fixed quantizer scale. For P- pictures, the estimated bandwidth can be obtained by encoding its coefficients after motion compensation prediction, or it can be obtained by using the estimated bandwidth of the previous P- picture. For B- picture, the bandwidth can be estimated by encoding the coefficients, or it can be estimated by using either the bandwidth of the previous B- picture or using universal constants from the previous P- picture. The target picture bit-rate is then determined by the ratio of the sum of reference bandwidths of all video streams to the sum of estimated bandwidths of all video streams. i.e.,

$$BW_k = \frac{\sum_{i=1}^{G} BW_{PCT_i}}{\sum_{i=1}^{G} BW_{EST_i}} \times BW_{EST_k} \qquad (2.8)$$

where, G is the number of video streams to be multiplexed.

As in the case of bit allocation for a single video source, the above techniques suffer from picture quality inconsistency when the picture characteristics are estimated from past statistics. Another problem of these techniques is that they treat the bit allocation problem at the picture level. That is, one picture from each video stream is processed one at a time. Treating the bit allocation at the picture level requires the re-computation of picture target at every picture. It also requires the distinction between I-, P-, and B-pictures. In [18], the picture coding types from all video programs are assumed to be synchronous. That is, only I-, P-, or B- pictures from all video streams are being processed at each instance. This assumption is rather limiting since no one can guarantee that at each instance, only one picture coding type is encountered. In [17], the distinction between I-, P-, and B- pictures is eliminated by assuming the same complexity for each picture in a scene. This assumption is usually invalid since P- and B- pictures utilize motion compensated prediction and thus, have lower bit requirements and lower picture complexities than those of the I- pictures. The technique found in [19] recognizes the necessity to distinguish between I-, P-, and B- pictures and provides different methods for finding the reference bandwidth and the estimated bandwidth of I-, P-, and B- pictures. The problem of picture coding type distinction can also be treated if the bit allocation decisions are performed at a higher level in the MPEG hierarchical structure than the picture level. An example of the latter approach is found in [16]. By performing bit allocation at the GOP level, this technique eliminates both the computation of the picture target for every picture and the necessary distinctions between I-, P-, and B- pictures when the bit allocation decisions are made at the picture level.

2.3 Summary

In this chapter we reviewed the basics of MPEG-2 video coding. An MPEG-2 video stream has six layers of information. The top level is the sequence layer, followed by the GOP layer, the picture layer, the slice layer, the macroblock layer, and the block layer. Compression in MPEG is partially achieved by the quantization of DCT transformed blocks. Two classes of video streams, CBR and VBR, can result from MPEG-2 video coding. VBR video streams, whose bit-rate is not fixed, offer more consistent picture quality than CBR video streams. The existing MPEG-2 video coding methods try to maintain consistency in picture quality via bit allocation processes. In bit allocation, the complexity of a picture is determined and a mapping function is applied to this picture complexity in order to obtain a picture target for the picture. Many existing techniques use past statistics to estimate picture complexities. Since video streams have non-stationary characteristics, the estimated picture complexities in many cases do not reflect the activity level of the current picture. In situations where a high channel bandwidth is available, multiple video sources can be multiplexed together for transmission. Statistic multiplexing and joint bit-rate control are two classes of bit allocation techniques used in multiple-source coding. In statistical multiplexing, video stream data are segmented into cells, and the cells are placed in a buffer. Immediately before transmission, cells from all sources are extracted from the buffer and multiplexed. Since video streams in statistical multiplexing are subject to cell loss, the channel is not used to its full capacity in order to avoid buffer overflows and long delays, which are the major contributors to cell loss. Joint bit-rate control allows the bit-rate of each video stream to vary according to the stream's activity level, while the sum of all bit-rates remains constant. Bit allocation in joint bit-rate control is done by determining the current picture complexity of each video stream and then distributing the channel bandwidth to each video stream according to the determined picture complexities. Many existing joint bit-rate control techniques estimate the picture complexities from past statistics. In addition, they perform bit allocation on a picture basis. Treating the bit allocation problem at the picture level requires the computation of picture target for every picture. It also requires distinctions between I-, P-, and B- pictures.

Chapter 3 Two-Stage Joint Bit-Rate Coding

Advancements in multimedia technology and digital communications enabled broadcasting of multiple programs in a channel, which was used to transmit a single analog program. Besides news, sports, and some other live shows, the majority of the broadcast programs such as movies, commercials, and music videos, are pre-recorded. The current practice in digital broadcasting is that broadcasters pre-record these materials into some storage medium such as tape. When it is time to transmit the programs, the contents are pulled out, encoded, multiplexed, and transmitted. The trade-off between picture quality and bandwidth is of the most importance in analog and digital broadcasting. The level of picture quality and the efficiency in bandwidth usage are strongly influenced by:

1. the quality of the video encoder,
2. the choice of the multiplexing method and the ability to optimally distribute bandwidth to the video streams based on content and complexity,
3. the available bandwidth,
4. and the number of programs to be transmitted simultaneously.

We develop a two-stage joint bit-rate coding system, which is a two-stage digital video encoding solution to multi-program transmission. This system guarantees that, given the available bandwidth, each individual video stream is assigned a desired bit-rate, and each individual picture within a video stream offers desired picture quality that reflects the content of the picture.

Acceptable picture quality can be achieved if a video encoding system is able to perform bit allocation that reflects the activity and complexity of the video stream. Many encoders use a global complexity measure, which is estimated from past statistics, to allocate a certain number of bits for each picture in a GOP. The ability of past statistics to make assumptions about the complexity of current pictures is limited. A good approximation can be made from past statistics only if the scene of the current picture is closely related to the picture from which the statistics are extracted. Unfortunately, given the non-stationary characteristics of video programs, this assumption is not always valid. Therefore, other measures are needed to determine the activity and complexity of a video stream. In our proposed two-stage joint bit-rate coding system, the picture quality issue is addressed via a two-stage encoding approach. In the first stage, the GOP and picture complexity statistics are found for each GOP. This information is then used on the same GOP in the second stage.

Guaranteed picture quality can be achieved if a multiplexing scheme does not under-assign bit-rates to each video stream. Similarly, efficient use of bandwidth can be achieved if the multiplexing scheme does not over-assign bit-rates to each video stream. For example, in CBR transmission, channel bandwidth is wasted when the relatively simple content is encoded with a high bit-rate. On the other hand, picture quality suffers when active content is encoded with a low bit-rate. In our proposed two-stage joint bit-rate coding system, the bandwidth issue is addressed via a joint bit-rate control scheme.

Chapter 3 Two-Stage Joint Bit-Rate Coding

This chapter is dedicated to describing the system we developed. In Section 3.1 we give an overview on our two-stage joint bit-rate coding system. In Section 3.2, we discuss the first-stage video encoding process in details. In Section 3.3, we describe our joint bit-rate control process. In Section 3.4, we introduce the concept of transcoding and describe how transcoding works in our solution to give the desired video bit-rates. Finally, we provide a summary in Section 3.5.

3.1 An overview of the Two-Stage Joint Bit-Rate Coding System

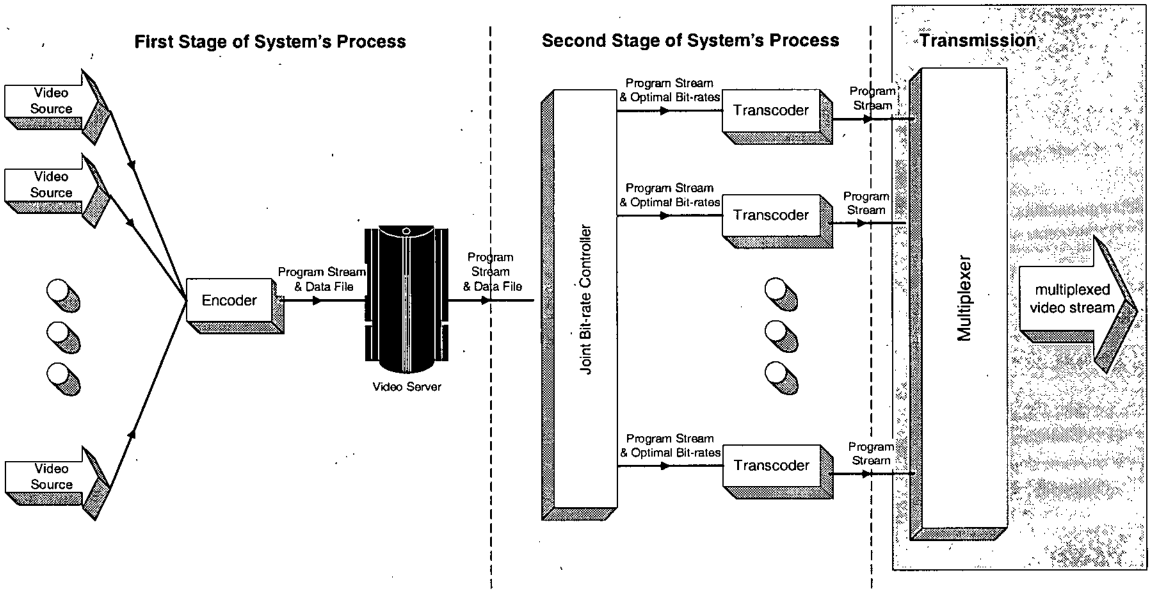

First Stage of System's Process
Second Stage of System's Process
Transmission
Video Source
Video Source
Video Source
Encoder
Program Stream & Data File
Video Server
Program Stream & Data File
Joint Bit-rate Controller
Program Stream & Optimal Bit-rates
Transcoder
Program Stream
Multiplexer
multiplexed video stream

Figure 3.1: Block diagram of the two-stage joint bit-rate coding system.

Figure 3.1 shows the block diagram of the proposed two-stage joint bit-rate coding system. The system has three key components: a video encoder and a video server, a joint bit-rate controller, and a set of transcoders. Generally, a transcoder is a mechanism that converts compressed video from one format to another [20]. In our system, the transcoder is a mechanism that converts a compressed video stream having a certain bit-rate to another having a different bit-rate. Our coding system divides the multi-program encoding process into two stages. The first stage involves an off-line encoding of the original video sources, using minimal compression and maintaining high picture quality. At this stage, the required statistics about the complexity of the different pictures in each video source are gathered. The outputs of this process consists of streams encoded in MPEG-2 format as well as data files containing the statistics of the video contents, which we call ***complexity files***. Both the encoded streams and their corresponding complexity files are stored in the video server. The encoded video streams that result from the first stage encoding process have a variable bit-rate, which is higher than that required for broadcasting. During the second stage, GOP targets for each video stream are determined, given the available bandwidth and the video content statistics. The joint bit-rate controller uses two complexity measures to guide the bit allocation decisions. Then, the high bit-rate video streams are transcoded using these GOP targets. The bit-rate conversion by the transcoder matches the encoded video streams of the first stage to the specifications and requirements of the broadcasting network. The resulting outputs of the system are video streams that are encoded with desirable picture quality and bit-rates.

The advantages of our two-stage joint bit-rate coding system over today's digital broadcast systems are fourfold:

1. more video streams are carried in a channel;
2. the picture quality of the video streams is more consistent;
3. the cost at the headend is lower.
4. the coding time of the video streams before transmission is shorter

While present broadcast systems requires one encoder for each video source for encoding the video sources simultaneously, video sources can be encoded one stream at a time in an off-line encoding process. Therefore, off-line encoding allows fewer complete encoders to be used. Video content analysis performed in the first stage produces statistics that reflect the activity and complexity of the video sources. Using these statistics to guide the bit allocation decisions results in video streams with desirable picture quality and desirable bit-rates. More video streams are carried in a channel as each video stream is assigned with only the necessary number of bits. During the first stage of our process, the computationally intensive motion compensation predictions are also performed. Of all the function blocks involved in an encoding process, motion compensation prediction accounts for most of the computation time. As an example, the two-step full-search block-matching algorithm used in an MPEG-2 encoder requires 90% ~ 95% of the total number of computations [22]. Thus, computing motion compensation prediction during the first stage reduces the complexity of a transcoder since the need for computing motion compensation prediction during the second stage is eliminated. This also reduces the encoding delay before transmission. Since fewer complete video encoders are required and the cost of a transcoder is significantly cheaper than a complete encoder, the cost at the headend is lowered.

3.2 First Stage Encoder

Video coding is the first stage of the two-stage joint bit-rate coding process. The objective of this stage is to encode the original video sources at very high variable bit-rates in order to guarantee video quality very close to the original. At the same time, complexity files, which reflect the complexity of the content of each video stream, are generated. Figure 3.2 illustrates the block diagram of the encoder. The encoder performs spatial transformation, fixed quantization, motion compensation prediction, and variable length coding on video sources. This encoder differs from the general MPEG-2 encoder in that fixed quantization is used; i.e., a single quantization step-size is utilized throughout the entire encoding process to quantize a video program.

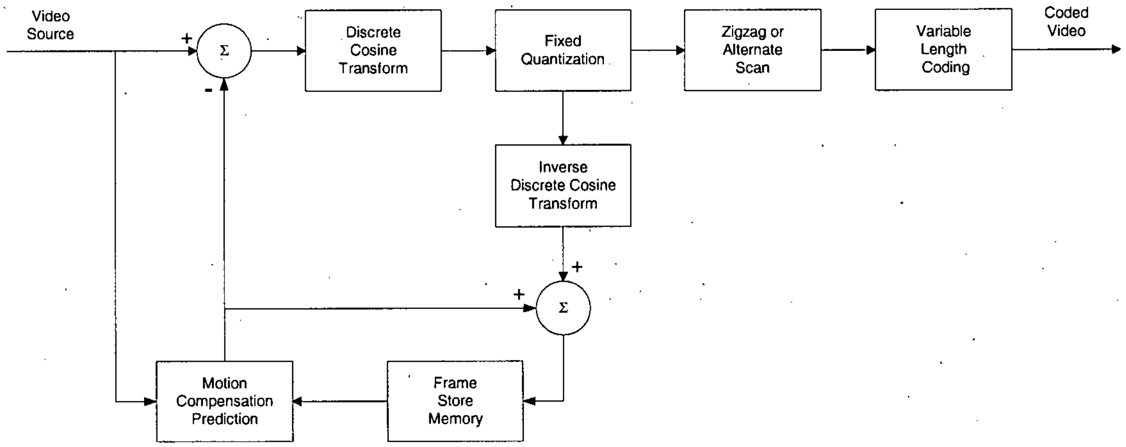


Figure 3.2: Block diagram of the first-stage video encoder.

3.2.1 Fixed Quantization

The MPEG-2 Video Specification [2] has defined two default quantizer matrices, one for intra-coded macroblocks and the other for inter-coded macroblocks (see Figure 3.3). If one or both quantizer matrices are not included in the video stream, the default values are used.

$$\begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \qquad \begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix}$$

(a) Default intra quantizer matrix (b) Default non-intra quantizer matrix

Figure 3.3: Default quantizer matrices defined in the MPEG-2 Video Specification.

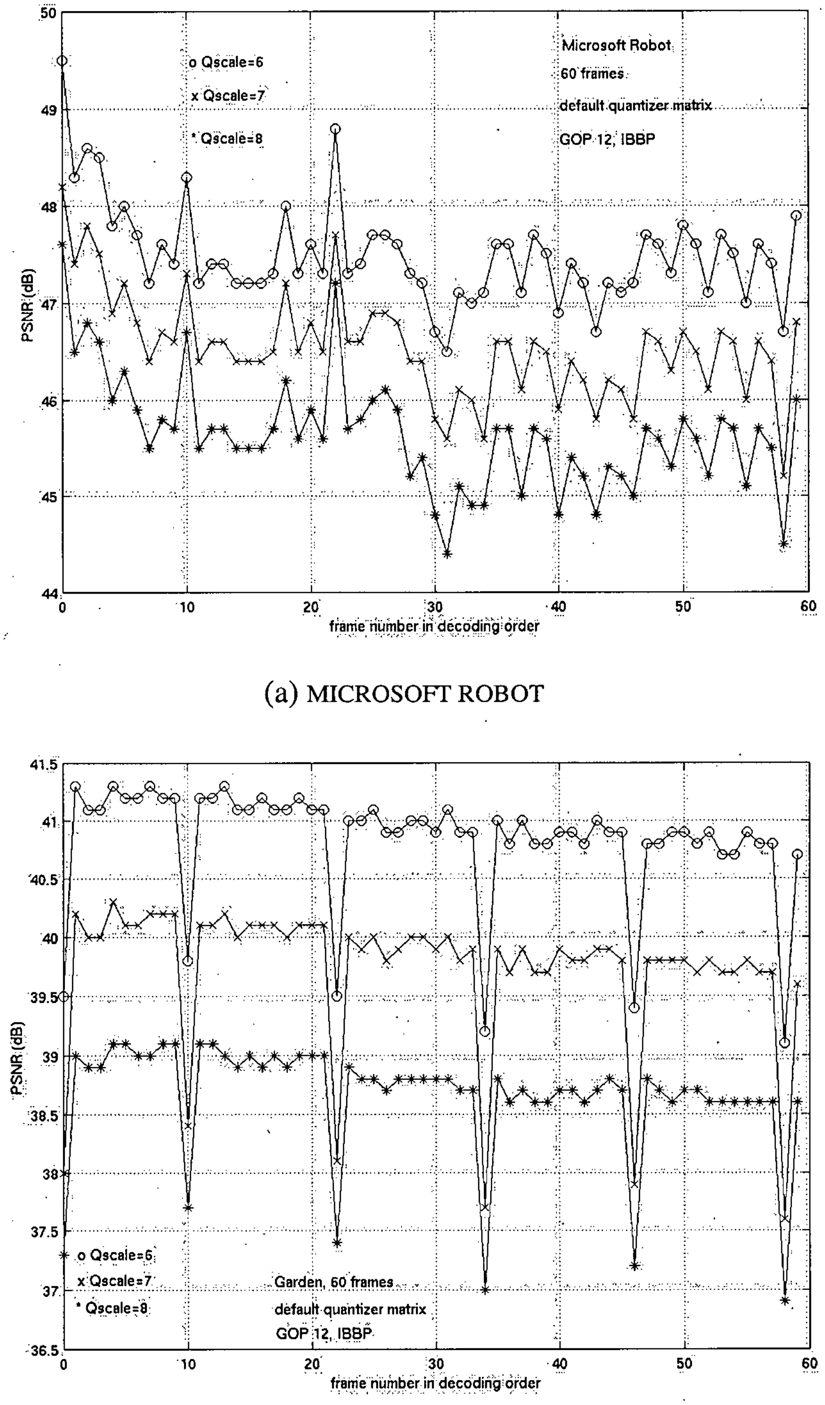
The majority of the weighting factors in the default intra quantizer matrix are larger than the weighting factors of the default non-intra quantizer matrix. Therefore, I-pictures, which are all intra-coded, have larger quantization step-sizes than P- and B-pictures when fixed quantization is used. Performance evaluations have shown that using the default matrices in the first stage of our process results in video quality significantly lower than that desired. Figure 3.4 shows the picture quality for two different video streams (MICROSOFT ROBOT and GARDEN) encoded using the default quantizer matrices. We use the peak signal-to-noise ratio (PSNR) of a picture as the picture quality. PNSR is defined as

$$PSNR = 10 \times \log_{10}\left(\frac{255}{MSE}\right) \qquad (3.1)$$

where, MSE is the mean squared error of a picture, which is computed as

$$MSE = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H}(org[i,j] - rec[i,j])^2 \qquad (3.2)$$

where, W is the horizontal resolution of the picture; H is the vertical resolution of the picture; org[i,j] and rec[i,j] are the pixel values of the original and reconstructed picture respectively.



(a) MICROSOFT ROBOT

(b) GARDEN

Figure 3.4: Picture quality (PSNR) of test sequences encoded using 3 fixed quantizer scales and the default quantizer matrices.

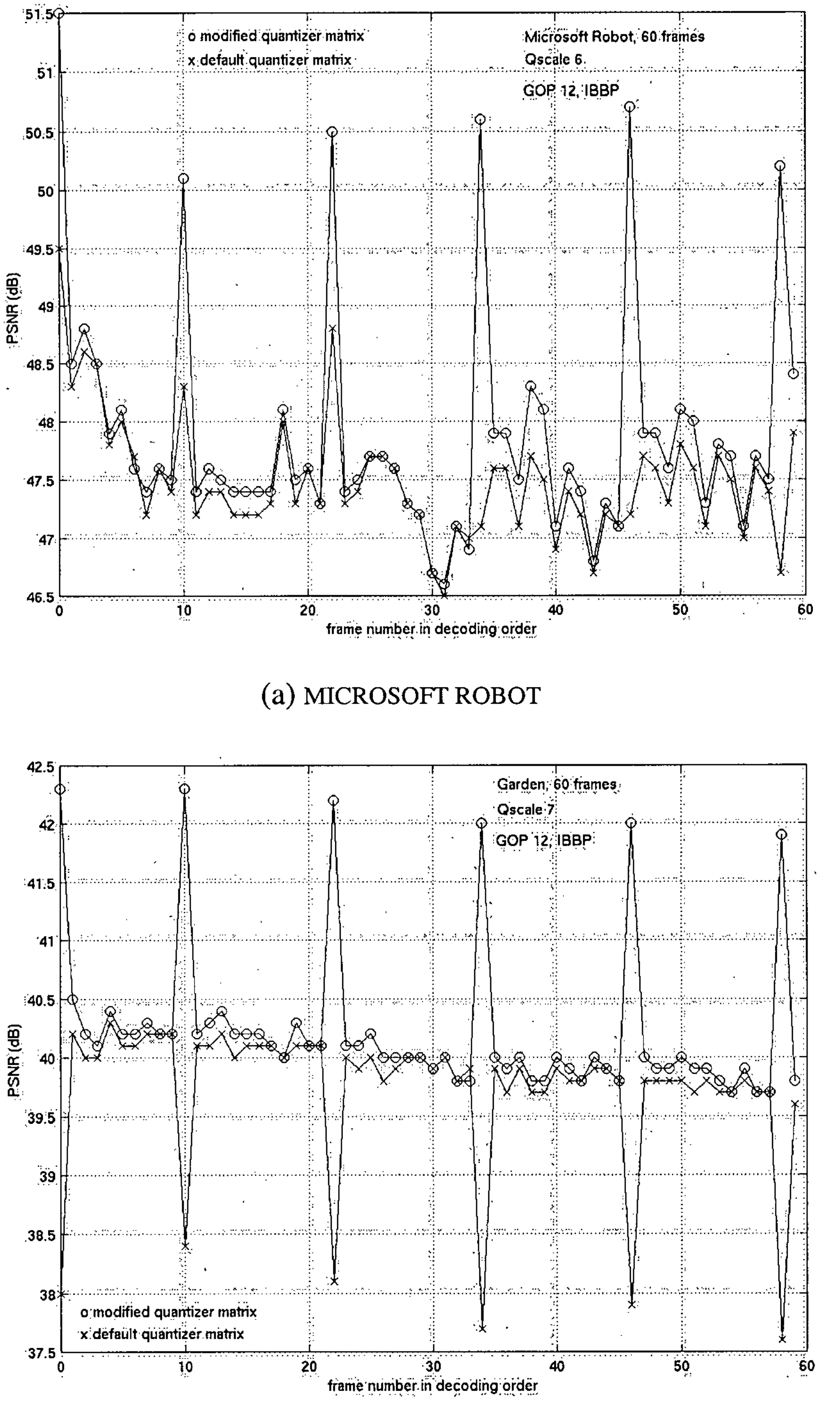
Since for our implementation, it is essential to maintain very high picture quality in the first stage video encoding, we propose the use of a modified intra quantizer matrix for intra-coded blocks during the first stage, which minimizes the distortion in the I- pictures. During the second stage, the default intra quantizer matrix will be used. For inter-coded blocks, the default non-intra quantizer matrix is used. Figure 3.5 shows the proposed modified intra quantizer matrix with weighting factors equal to those of the default non-intra quantizer matrix except for the DC term. The proposed smaller weight factors result in less distortion in the overall video stream. Table 3.1 shows the PSNR values obtained using the default intra quantizer matrix as well as our proposed matrix. We observe that the test sequences have an overall 0.5 dB increase in average PSNR, a 2.8 dB increase in the average PSNR of I- pictures for MICROSOFT ROBOT, and a 4.3 dB increase in the average PSNR of I- pictures for GARDEN. These figures translate to an average 5.4% and 11.3% improvements in I- pictures for the MICROSOFT ROBOT sequence and the GARDEN sequence respectively. Figure 3.6 compares the picture quality of the video streams obtained using the default and our proposed matrices. From the graphs, we observe that the use of our proposed matrix significantly improves the quality of the pictures.

$$\begin{bmatrix} 8 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix}$$

Figure 3.5: The modified intra quantizer matrix.

Matrix	Quantizer Scale	Average PSNR (dB)	Average PSNR (dB) I- picture	Average PSNR (dB) P- picture	Average PSNR (dB) B- picture
MICROSOFT ROBOT					
Default	6	47.49	47.93	47.25	47.52
Proposed	6	47.90	50.60	47.41	47.67
Default	7	46.54	46.63	46.35	46.60
Proposed	7	46.98	49.32	46.55	46.80
Default	8	45.64	45.98	45.37	45.68
Proposed	8	46.14	49.00	45.55	45.92
GARDEN					
Default	6	40.84	39.42	41.09	40.96
Proposed	6	41.32	43.73	41.19	41.01
Default	7	39.74	37.95	40.01	39.90
Proposed	7	40.24	42.12	40.13	39.98
Default	8	38.86	37.25	38.82	38.81
Proposed	8	39.21	41.73	38.97	38.92

Table 3.1: PSNR values of test sequences encoded with the default and proposed intra quantizer matrices.



(a) MICROSOFT ROBOT

(b) GARDEN

Figure 3.6: Picture quality (PSNR) comparisons of test sequences encoded with the default and the modified intra quantizer matrix.

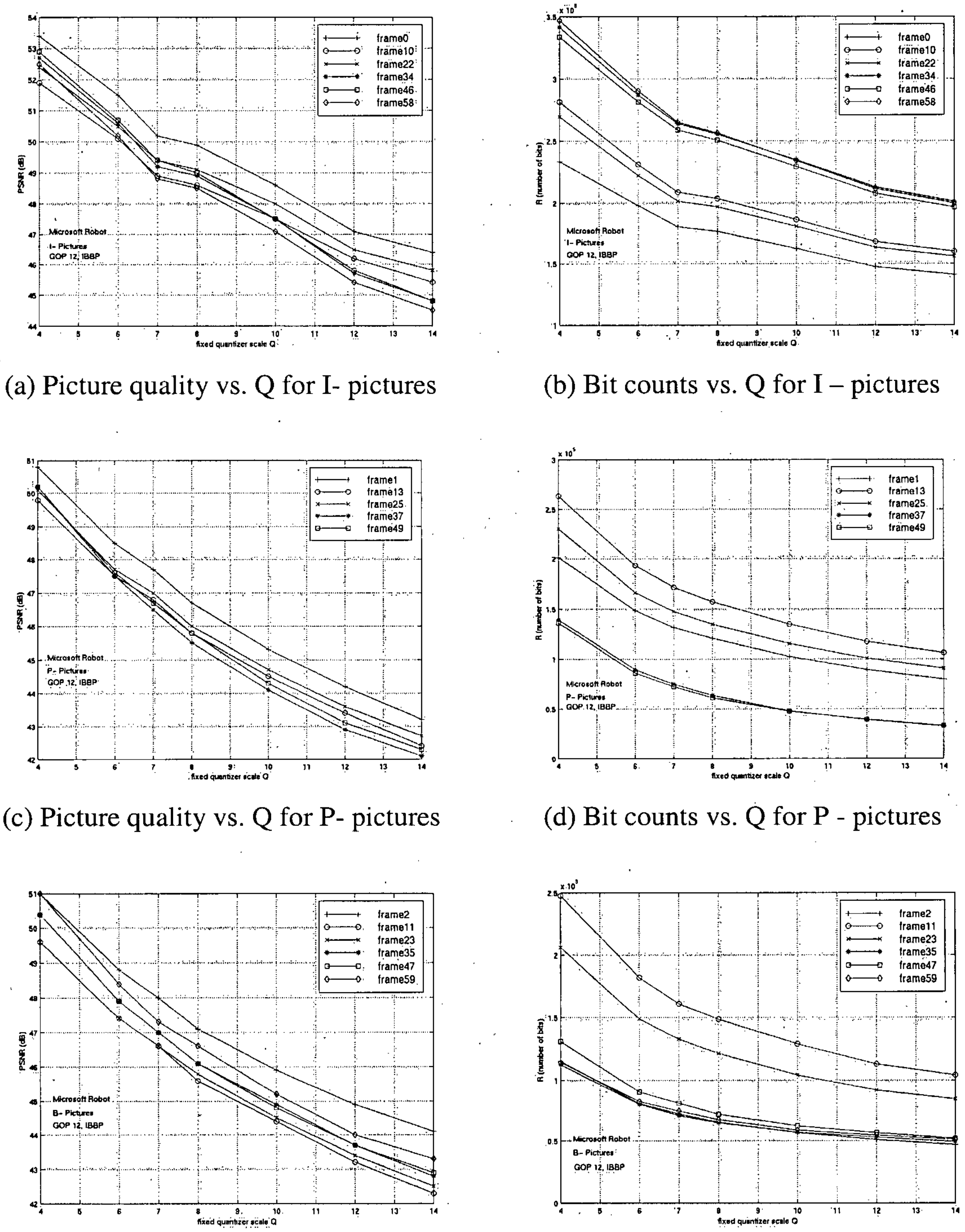
3.2.2 Video Server: Trade-off between Cost and Picture Quality

Video sources are usually represented in D1[1] format. The latter requires a huge number of bits to represent the video pictures. Although the price of disk storage has come down drastically over the past years, video servers are not cost effective in storing D1 quality videos. It is, thus, necessary to compress the original video sources before we store them in video servers. There is a trade-off between cost of storage and picture quality.

Since quantization is a lossy operation, the larger the quantization step-size used, the more distortion is introduced in a picture. However, it is also true that the larger the quantization step-size used in quantization, the smaller number of bits is needed to encode a picture. We determine a maximum fixed quantizer scale parameter, which guarantees no visible artifacts after transcoding the video streams. Although smaller quantizer scale may also be used, it is important to note that the same fixed quantizer scale parameter should be used for all video streams, in order to correctly compare them during the second stage of our process. Figure 3.7 and 3.8 show the quality-quantization (***Q-Q or PSNR-Q***) relationship and the rate-quantization (***R-Q***) relationship for the MICROSOFT ROBOT and GARDEN video streams. The PSNR-Q relationship describes how the picture quality of a picture changes as the quantizer scale changes. The R-Q relationship describes how the number of bits needed to encode a picture changes as a function of the quantizer scale. We observe that the decrease in the PSNR or the number of bits used to encode a picture gets smaller as Q increases. Performance evaluations have also shown that by using a quantizer scale greater than 7, artifacts of a video stream

[1] D1 refers to the non-compressed Standard Definition digital video signal with 4:2:2 format [27].

become visible after the video stream has been transcoded. Therefore, 7 is our maximum quantizer scale value.



(a) Picture quality vs. Q for I- pictures

(b) Bit counts vs. Q for I – pictures

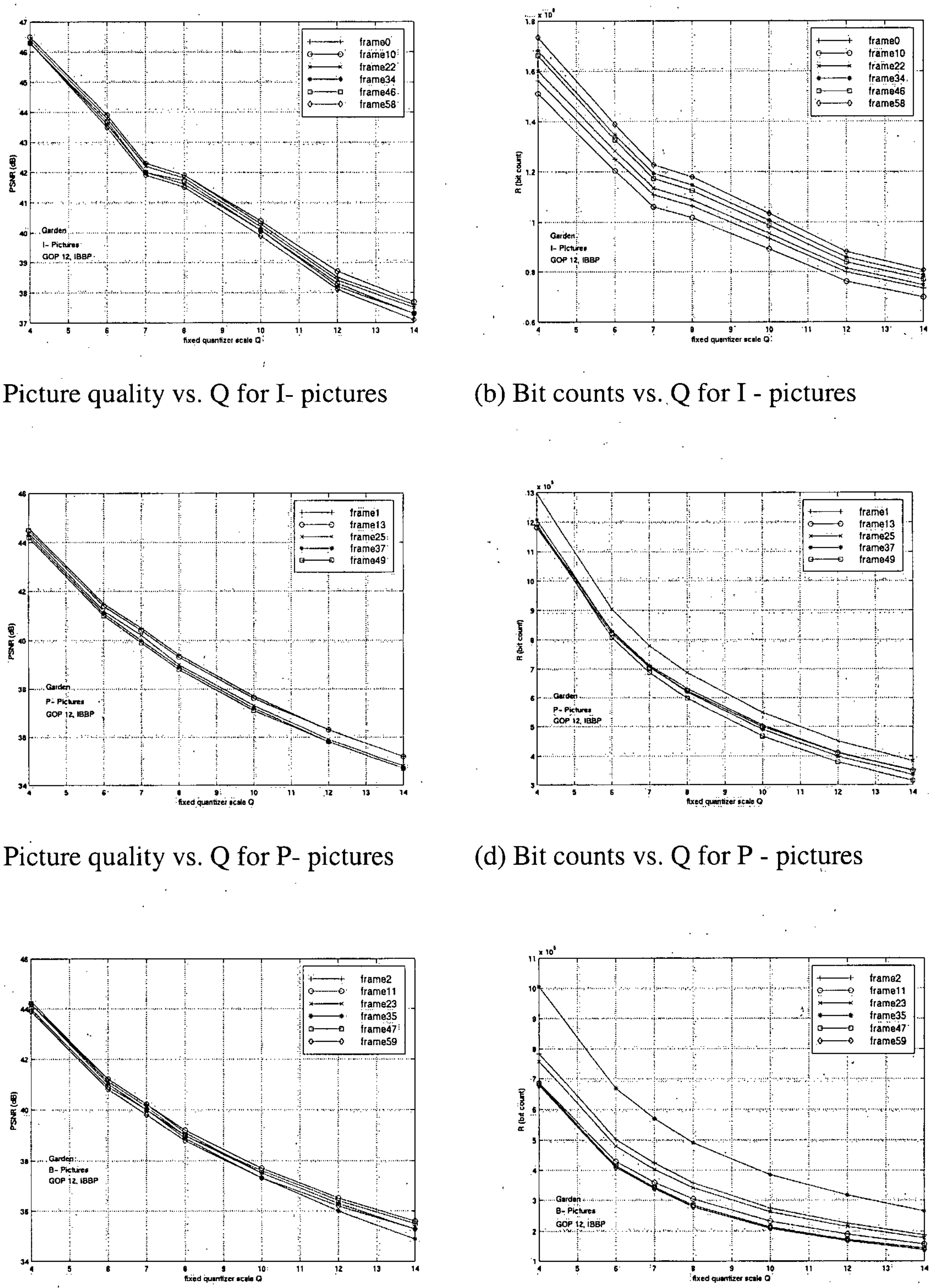
(c) Picture quality vs. Q for P- pictures

(d) Bit counts vs. Q for P - pictures

(e) Picture quality vs. Q for B- pictures

(f) Bit counts vs. Q for B - pictures

Figure 3.7: PSNR-Q and R-Q Curves of MICROSOFT ROBOT.



(a) Picture quality vs. Q for I- pictures

(b) Bit counts vs. Q for I - pictures

(c) Picture quality vs. Q for P- pictures

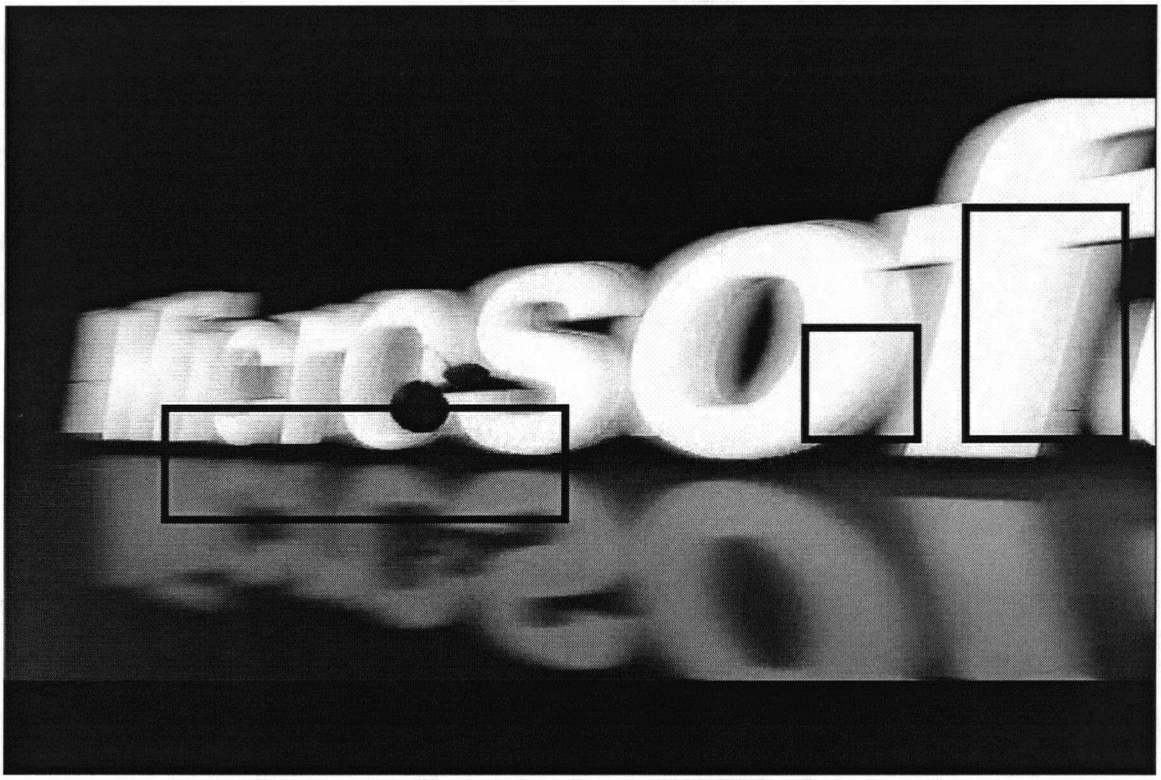
(d) Bit counts vs. Q for P - pictures

(e) Picture quality vs. Q for B- pictures

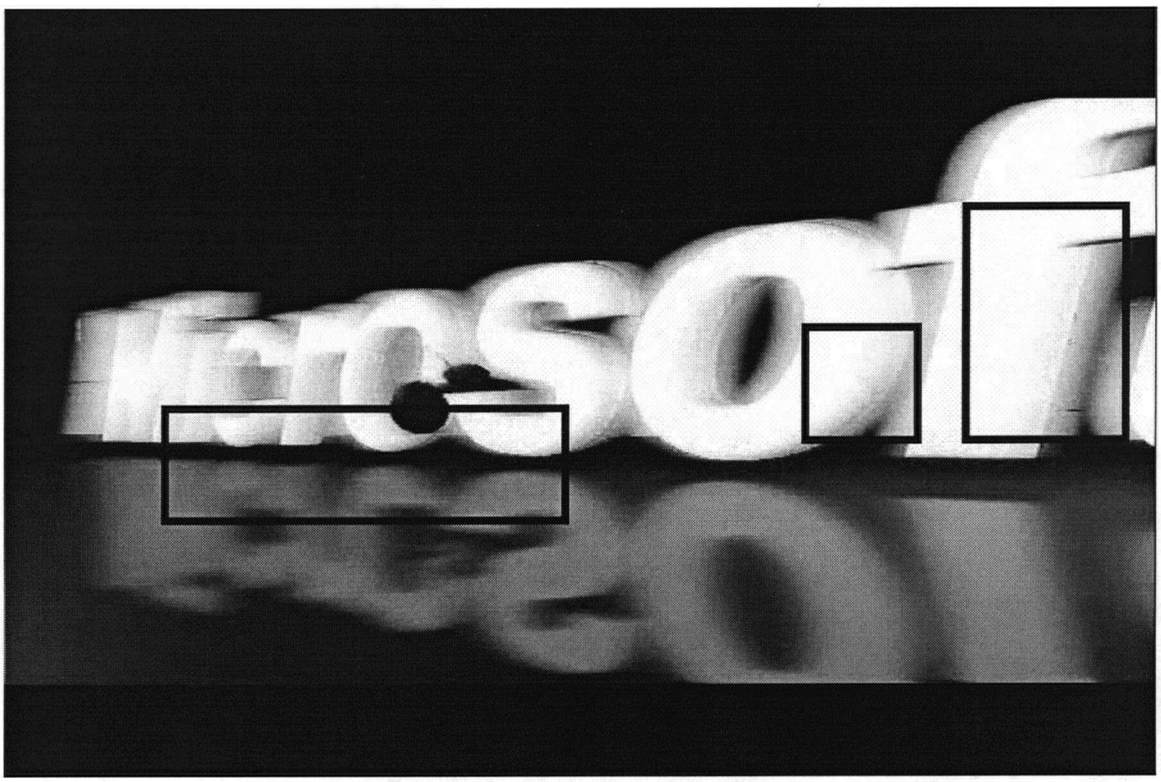
(f) Bit counts vs. Q for B - pictures

Figure 3.8: PSNR-Q and R-Q Curves of GARDEN.

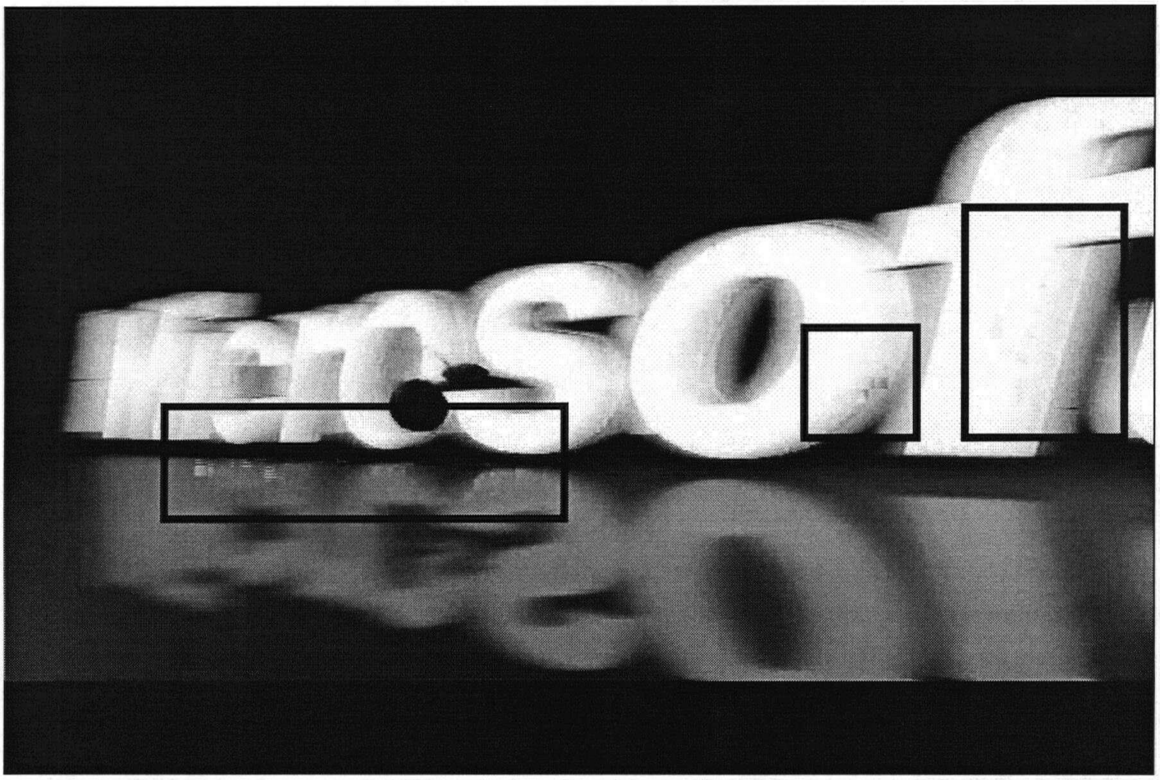
Figure 3.9 and 3.10 compare the picture quality of two video streams (MICROSOFT ROBOT and GARDEN) that have been encoded with fixed quantizer scales 7, 8, and 10 and transcoded to the same variable bit-rates. Both video streams are 2 seconds in length and have a pixel size 720x480. The video streams have been encoded at 30 frames/sec, with a GOP size of 12, and a GOP pattern of IBBPBBPBBPBB. The variable bit-rate assignments are obtained by putting the video streams through our two-stage joint bit-rate coding system. In Figure 3.9 and 3.10, the rectangles outline the areas where artifacts cannot be seen when the video streams are quantized with a quantizer scale of 7, but the artifacts become visible when the video streams are quantized with larger quantizer scales.



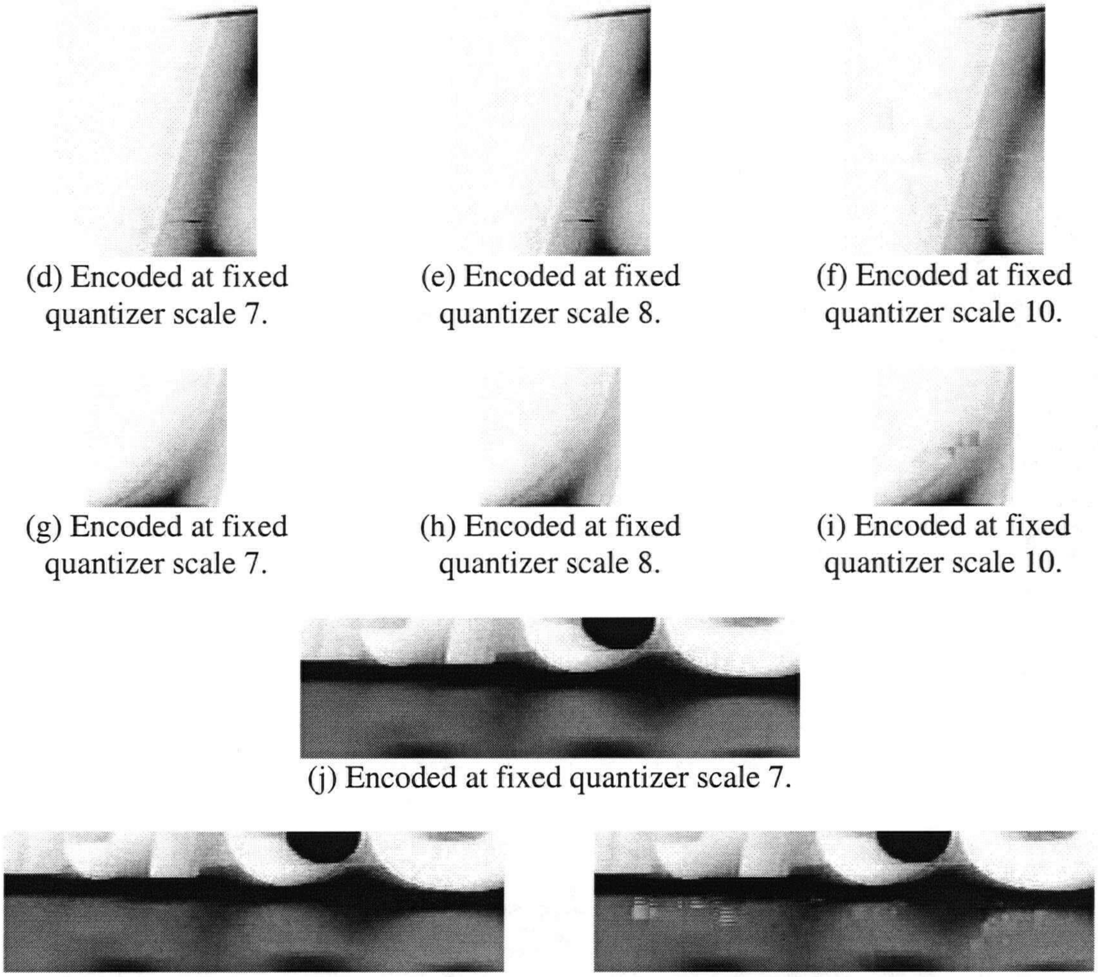
(a) Encoded fixed quantizer scale 7.



(b) Encoded at fixed quantizer scale 8.



(c) Encoded at fixed quantizer 10.



(d) Encoded at fixed quantizer scale 7.

(e) Encoded at fixed quantizer scale 8.

(f) Encoded at fixed quantizer scale 10.

(g) Encoded at fixed quantizer scale 7.

(h) Encoded at fixed quantizer scale 8.

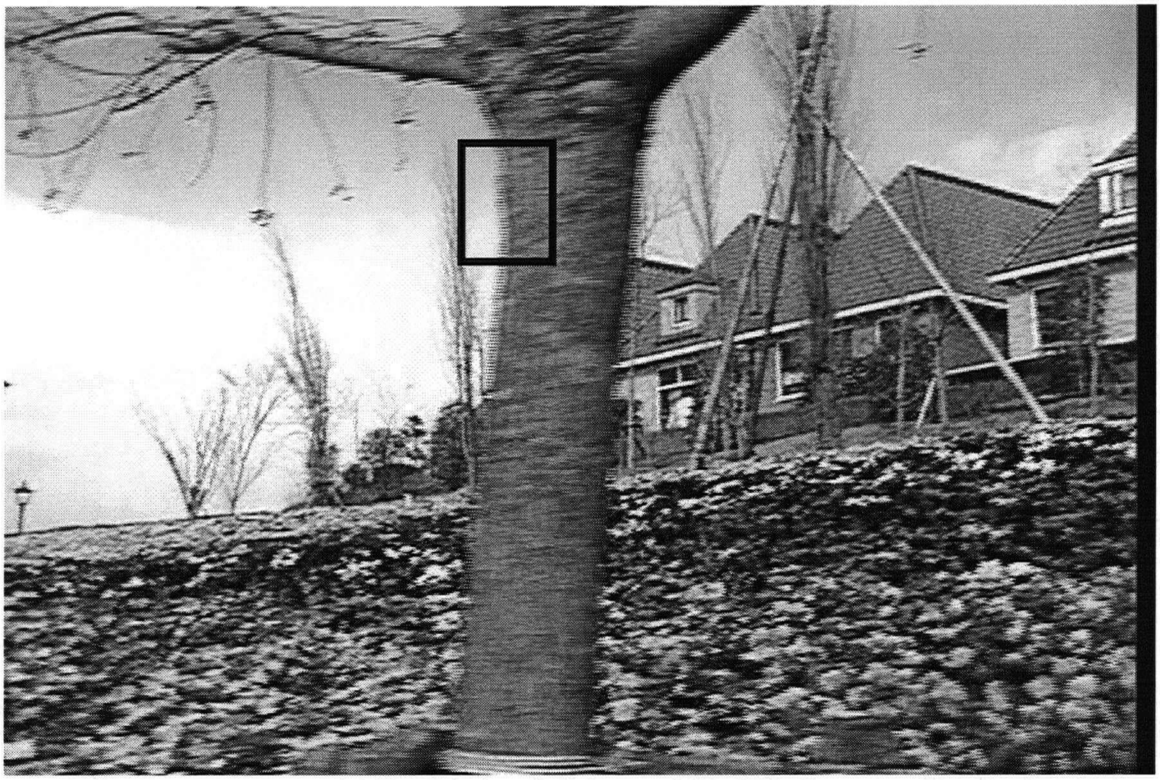
(i) Encoded at fixed quantizer scale 10.

(j) Encoded at fixed quantizer scale 7.

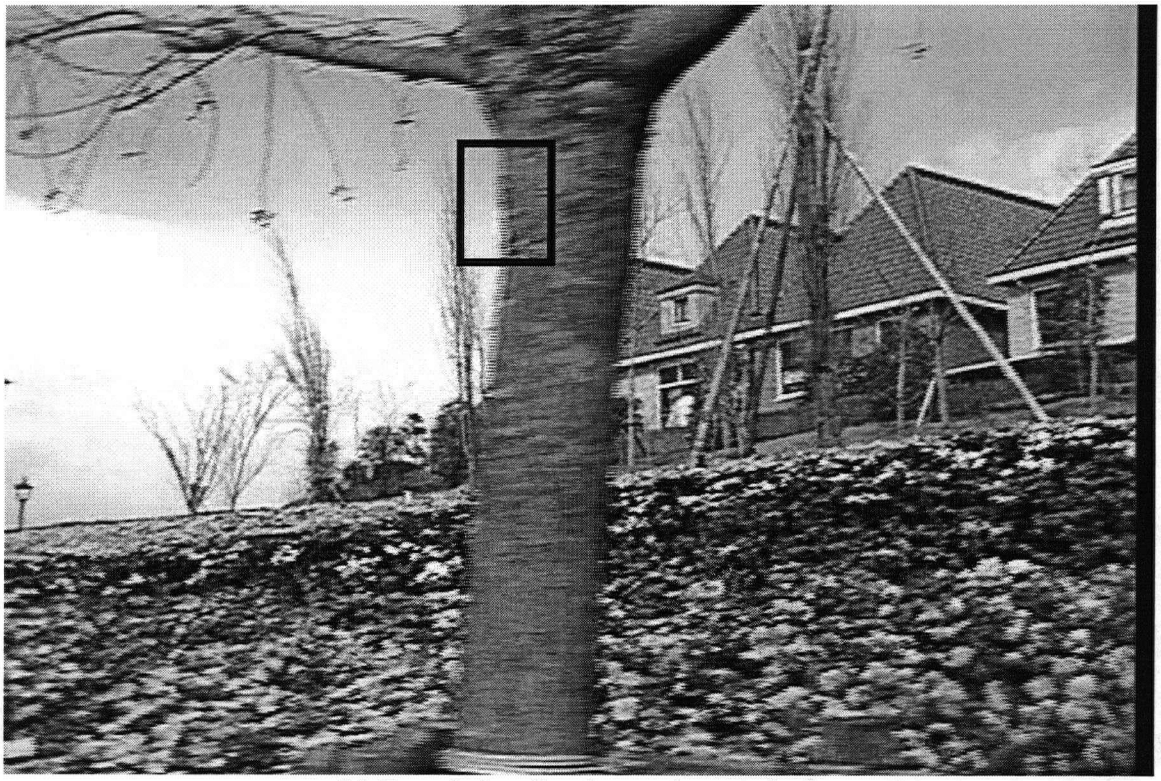
(k) Encoded at fixed quantizer scale 8.

(l) Encoded at fixed quantizer scale 10.

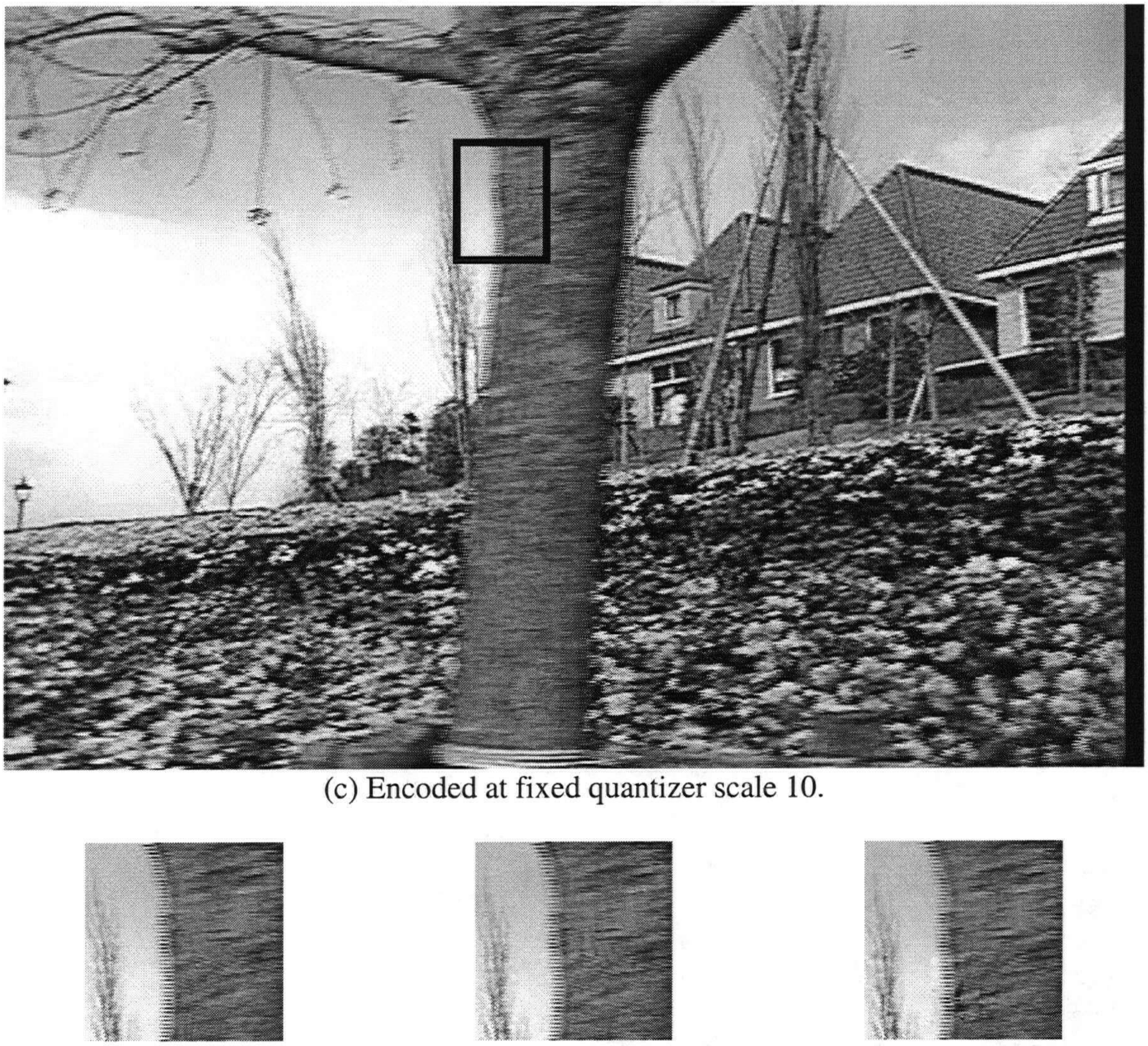
Figure 3.9: Picture quality comparisons among MICROSOFT ROBOT encoded at fixed quantizer scales 7, 8, and 10.



(a) Encoded at fixed quantizer scale 7.



(b) Encoded at fixed quantizer scale 8.



(c) Encoded at fixed quantizer scale 10.

(d) Encoded at fixed quantizer scale 7.

(e) Encoded at fixed quantizer scale 8.

(f) Encoded at fixed quantizer scale 10.

Figure 3.10: Picture quality comparisons among GARDEN encoded at fixed quantizer scales 7, 8, and 10.

3.2.3 Complexity Files

A complexity file is a record of the picture and GOP complexities. ***Picture complexity*** is defined as the number of bits used to encode a picture during the first stage of our process. The picture bit counts from this stage reflect the complexity of the picture content because a single quantizer scale value is applied to the entire video stream. Therefore, if a picture is highly active, its spatial transformation will have more non-zero coefficients than pictures which has low activity levels, and thus, more bits are needed to encode the picture. ***GOP complexity*** is defined as the sum of the picture complexities in a GOP, which is also the number of bits used to encode all pictures of a GOP. GOP complexities are used in joint bit-rate control to determine GOP targets. Picture complexities are used in the transcoding process to determine the picture bit distributions within a GOP. Besides picture complexities and GOP complexities, the following parameters are also included in a complexity file:

N - the number of frames within a GOP;

M - (M-1) is the number of consecutive B frames between an I frame and the first P frame following it (or between two consecutive P frames);

frame_rate_code - a value that represents the frame rate used in the video stream;

max_rate - a user-defined bit-rate that represents the bit-rate needed by the video stream during which the video content is most active;

min_rate - a user-defined bit-rate that represents the bit-rate needed by the video stream during which the video content is least active;

number_of_frames - the number of frames the video stream comprises.

3.3 Joint Bit-Rate Controller

The second stage of our encoding system consists of a joint bit-rate controller that oversees the bit allocation operations of the video streams to be multiplexed in a single channel. This controller receives the encoded video streams and their corresponding complexity files, which were generated during the first encoding stage. Based on this information and the given channel bandwidth, the controller determines a GOP target for each video source on a GOP basis and sends these GOP targets to the set of transcoders. The controller assumes that all video streams have the same N, M, and *frame_rate* or the same $\frac{frame\_rate}{N}\left(=\frac{\#\text{of GOP's}}{\text{sec}}\right)$. For a given bandwidth, the aim of our joint bit-rate controller is to offer appropriate bit allocation and consistent picture quality for each GOP of the video streams.

There are two functions performed by the joint bit-rate controller:

1. an admission test to accept or reject each video requesting to be transmitted and
2. a processing procedure for each GOP to determine the appropriate GOP target for each video source.

Figure 3.11 illustrates the dataflow diagram of the joint bit-rate control process.

3.3.1 Admission Test

The purpose of the admission test is to reject any new video stream requesting to be transmitted, if its addition to the system degrades the quality of the presently transmitted video programs to an unacceptable level. Our admission test follows the following simple procedure: It sums up the *min_rate* of all streams currently present in the system as well as the *min_rate* of the new stream. If the sum is greater than the channel rate, the new video is not accepted for transmission; otherwise, the new stream is added to the system. If a more sophisticated admission test is required, this module can be easily replaced or modified since replacement does not affect the operation of the system; the entire system would work the same way as before.

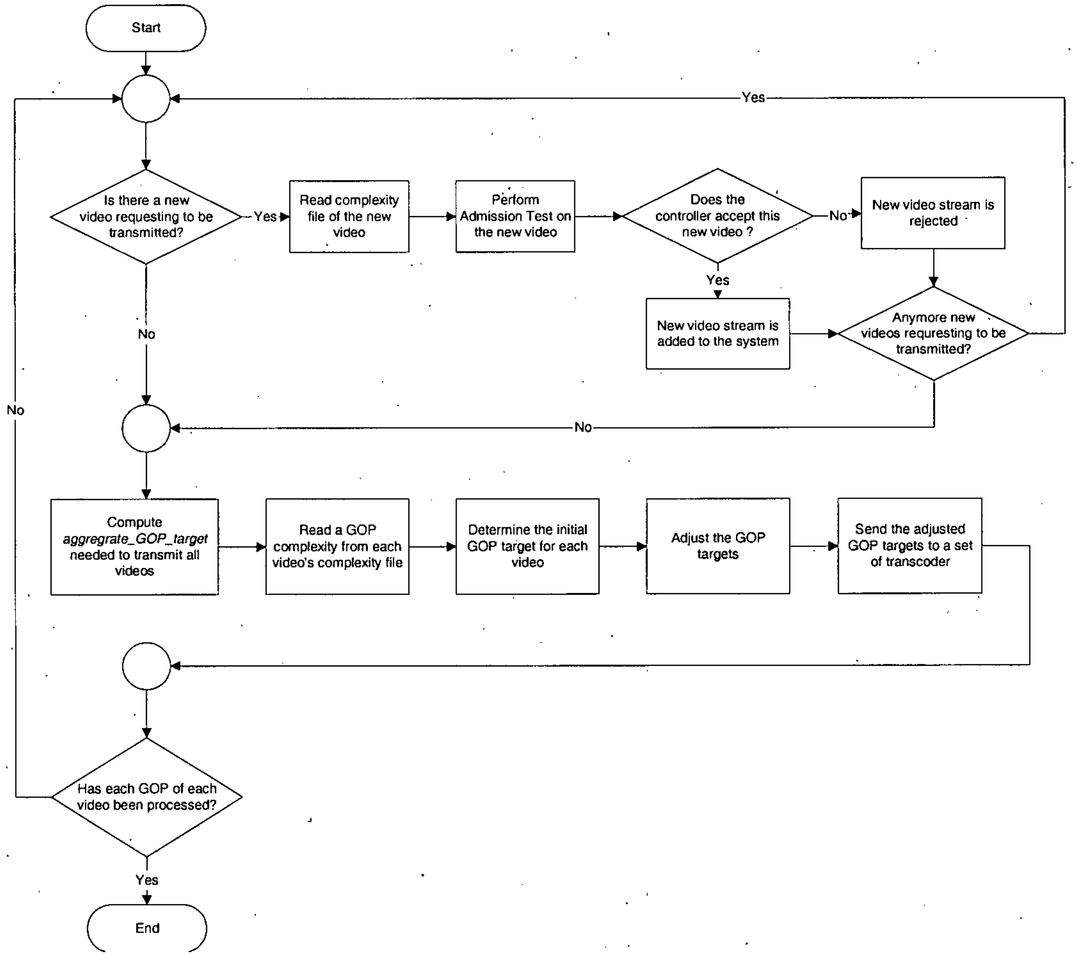

Start
Yes
Is there a new video requesting to be transmitted?
Yes
Read complexity file of the new video
Perform Admission Test on the new video
Does the controller accept this new video ?
No
New video stream is rejected
Yes
New video stream is added to the system
Anymore new videos requresting to be transmitted?
No
No
No
Compute *aggregrate_GOP_target* needed to transmit all videos
Read a GOP complexity from each video's complexity file
Determine the initial GOP target for each video
Adjust the GOP targets
Send the adjusted GOP targets to a set of transcoder
Has each GOP of each video been processed?
Yes
End

Figure 3.11: Dataflow diagram of the joint bit-rate control process.

3.3.2 Processing Procedure for a Group of Pictures

For every GOP, the joint bit-rate controller performs the following five steps:

1. it computes the necessary number of bits required to transmit the GOP's of all video streams, which we call *aggregrate_GOP_target*;
2. it reads the GOP complexity of the current GOP of each individual video stream;
3. it determines an initial GOP target of each video stream based on the square root of its complexity relative to the sum of the square root complexity of all video streams;
4. it adjusts the value of the initial GOP target of each video stream based on the user-defined *max_rate* and *min_rate*;
5. it sends to a set of transcoders the adjusted GOP target at which each corresponding video stream must be transcoded.

While the admission test ensures that each video stream present in the system has at least the minimum bandwidth it requires, the computation of the *aggregrate_GOP_target* ensures that all video streams together do not use more than the necessary bandwidth. This means, that the video streams do not always consume the entire channel bandwidth. The free bandwidth could be used to provide other services such as Internet and long-distance telephone. The *aggretrate_GOP_target* is determined as

$$\begin{aligned} &necessary\_bandwith = \min(channel\_bandwidth, \sum_{i=1}^{Np} max\_rate_i) \\ &aggregrate\_GOP\_target = necessary\_bandwidth / (\tfrac{\#\text{ of GOP's}}{\text{sec}}) \end{aligned} \qquad (3.3)$$

where, N_p is the number of video streams present in the system; $\frac{\#\text{of GOP's}}{\text{sec}}$ specifies the number of GOP's transmitted per second; the sum of *max_rates* indicates the maximum rate needed to transmit the current GOPs of all the video streams, assuming each of the current GOP's comprises the most active segment of its video stream. By using the minimum of this sum and the channel bandwidth, we eliminate the possibility of over-assigning bandwidth to the video streams.

The joint bit-rate controller allocates a fraction of the *aggregrate_GOP_target* to the current GOP of each video stream. The initial fraction assigned to each video stream is proportional to the relative GOP complexity of the video streams. That is,

$$GOP\_target_i = \frac{\sqrt{GOP\_Complexity_i}}{\sum_{k=1}^{Np}\sqrt{GOP\_Complexity_k}} * aggregrate\_GOP\_target \qquad (3.4)$$

The square root function is applied to the GOP complexities because it compresses the complexity ratio between the high-complexity streams and the low-complexity streams into a reasonable range of values. When the complexities of the video streams reach extreme high levels, using the ratio $\frac{GOP\_Complexity}{\sum_{k=1}^{Np} GOP\_Complexity_k}$ results in allocating a large proportion of the *aggregrate_GOP_target* to the high-complexity streams, leaving insufficient number of bits for the low-complexity streams. Performance evaluations have shown that assigning GOP targets according to $\frac{\sqrt{GOP\_Complexity}}{\sum_{k=1}^{Np}\sqrt{GOP\_Complexity_k}}$ improves the bit allocation between different sources.

A special case for the joint bit-rate control algorithm is when there is only one video stream present in the system. When this situation occurs, the joint bit-rate controller will assign the same GOP target for all GOP's. As a consequence, the resulting video stream after transcoding is a CBR video stream.

In some cases, the assigned GOP target may go beyond the $max\_target\ \left(= max\_rate/(\frac{\#\text{ of GOP's}}{\text{sec}})\right)$ or below the $min\_target\ \left(= minrate/(\frac{\#\text{ of GOP's}}{\text{sec}})\right)$ derived from the user-defined *max_rate* and *min_rate*. This is especially true when some video streams present in the system have extremely high GOP complexities while others have extremely low GOP complexities. The *aggregrate_GOP_target* is determined using the *max_rate* of each video stream and distributed among the video streams according to their present GOP complexities. If low-complexity segments of the video streams are encountered, the controller will assign to those video streams GOP targets lower than their corresponding *max_target*. The video streams currently having high-complexity segments, on the other hand, will be given GOP targets above their respective *max_target*. Therefore, some GOP target adjustment is necessary. Figure 3.12 shows the flow diagram of the GOP target adjustment algorithm we applied after the initial GOP target assignment.

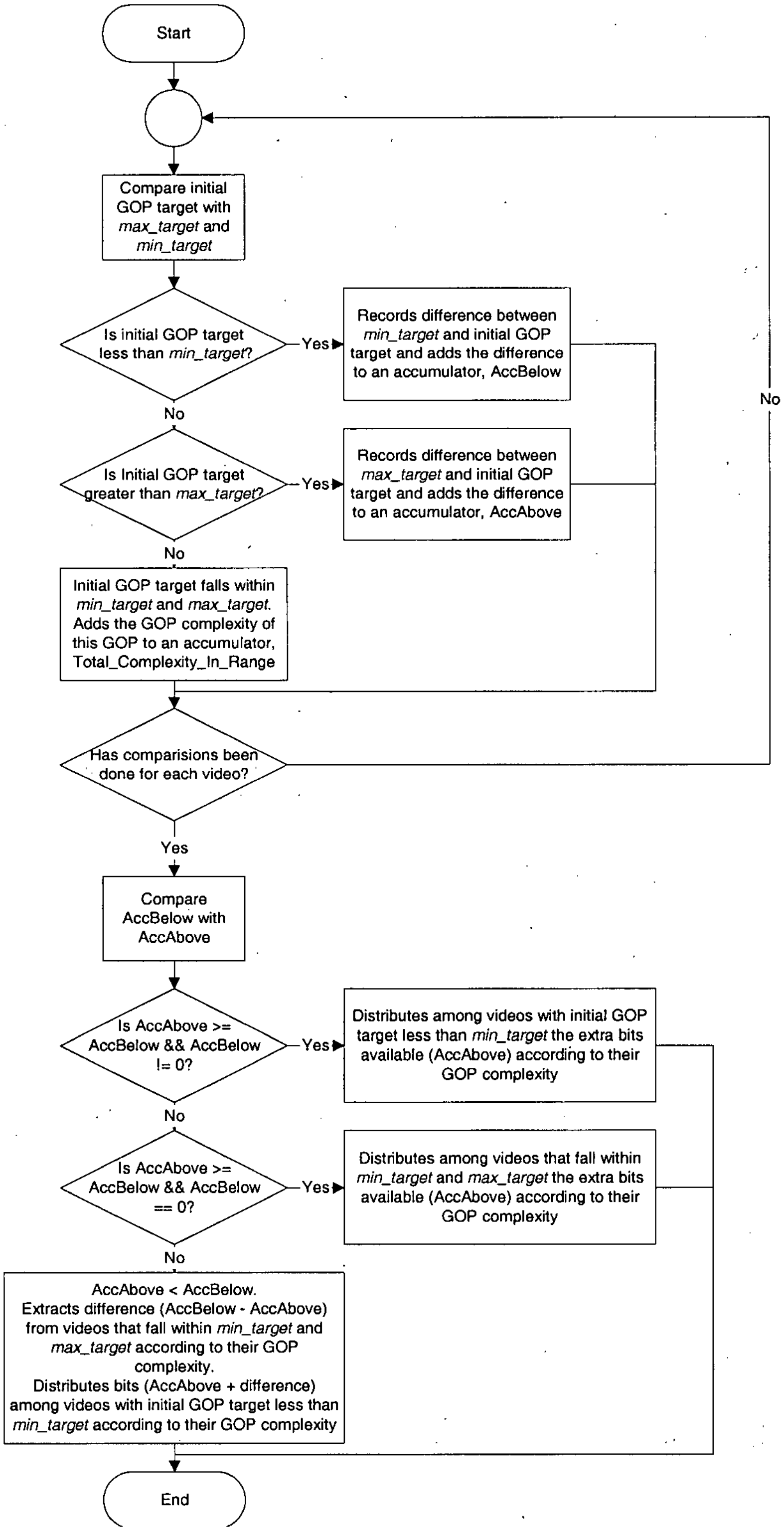

Start
Compare initial GOP target with *max_target* and *min_target*
Is initial GOP target less than *min_target*?
Yes
Records difference between *min_target* and initial GOP target and adds the difference to an accumulator, AccBelow
No
Is Initial GOP target greater than *max_target*?
Yes
Records difference between *max_target* and initial GOP target and adds the difference to an accumulator, AccAbove
No
Initial GOP target falls within *min_target* and *max_target*. Adds the GOP complexity of this GOP to an accumulator, Total_Complexity_In_Range
Has comparisions been done for each video?
No
Yes
Compare AccBelow with AccAbove
Is AccAbove >= AccBelow && AccBelow != 0?
Yes
Distributes among videos with initial GOP target less than *min_target* the extra bits available (AccAbove) according to their GOP complexity
No
Is AccAbove >= AccBelow && AccBelow == 0?
Yes
Distributes among videos that fall within *min_target* and *max_target* the extra bits available (AccAbove) according to their GOP complexity
No
AccAbove < AccBelow. Extracts difference (AccBelow - AccAbove) from videos that fall within *min_target* and *max_target* according to their GOP complexity. Distributes bits (AccAbove + difference) among videos with initial GOP target less than *min_target* according to their GOP complexity
End

Figure 3.12: Flow diagram of GOP target adjustment algorithm.

Basically, the GOP target adjustment algorithm compares the initial GOP target of each video with the video's *min_target* and *max_target*. One of three cases arises as the result of the comparison:

1. The initial GOP target is below *min_target*. The algorithm records the difference between *min_target* and the initial GOP target and adds the difference to an accumulator, AccBelow. The difference between *min_target* and the initial GOP target comprises the additional number of bits that the video stream must have for this GOP in order to achieve the minimum acceptable video quality.
2. The initial GOP target is greater than *max_target*. The algorithm records the difference between *max_target* and the initial GOP target and adds the difference to an accumulator, AccAbove. The difference between *max_target* and the initial GOP target represents the allocated number of bits that can be reduced from the GOP while still meeting the user's requirement.
3. The initial GOP target falls within *min_target* and *max_target*. The algorithm adds the GOP complexity of this GOP to an accumulator, *Total_Complexity_In_Range*, for later use.

Next, the bit adjustment algorithm compares AccBelow with AccAbove. One of three cases arises:

1. AccAbove is greater than AccBelow and AccBelow is not zero. That is, the extra number of bits obtained from the video streams that have initial GOP targets greater than their *max_targets* can totally compensate for the extra bits required by the video streams that have initial GOP targets less than their *min_targets*. Thus, the algorithm

distributes the extra bits among the videos with initial GOP targets below *min_target*. The extra bits are distributed as follows:

$$extra\_bits_i = \frac{GOP\_Complexity_i}{\sum_{i=1}^{N_{below}} GOP\_Complexity} * extra\_bits \qquad (3.5)$$

where, N_{below} is the number of videos that have their initial GOP bit-rate below their *min_rate*.

2. AccAbove is greater than AccBelow and AccBelow is zero. That is, there are extra bits available, but no streams are starved for bits. The algorithm, in this case, distributes the extra bits to the video streams with initial GOP targets falling within the *min_target*, *max_target* range, improving their video quality. The extra bits are distributed as follows:

$$extra\_bits_i = \frac{GOP\_Complexity_i}{\sum_{i=1}^{N_{in\_range}} GOP\_Complexity} * extra\_bits \qquad (3.6)$$

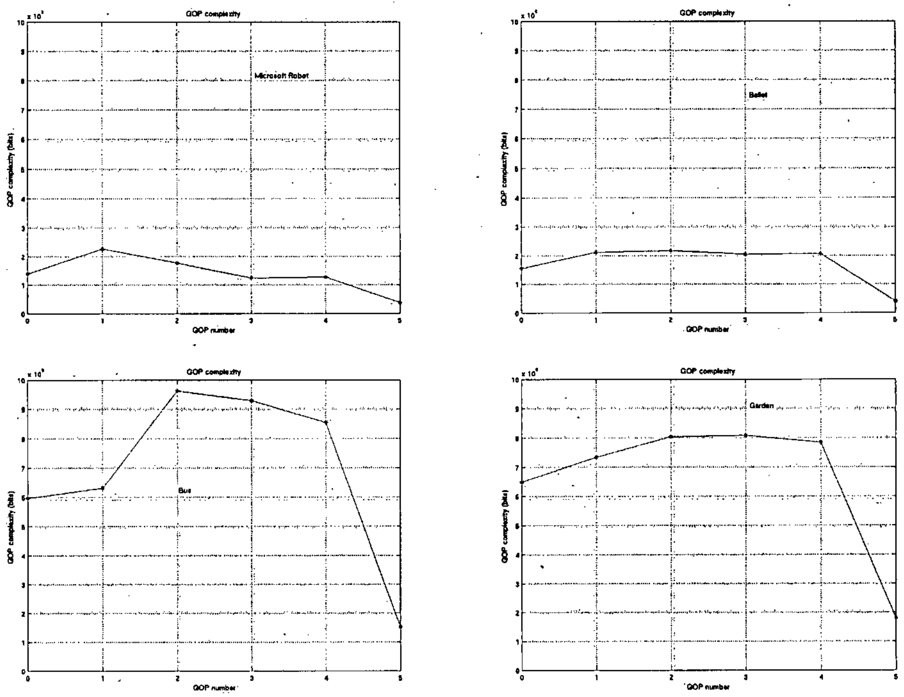
where, N_{in_range} is the number of videos whose initial GOP bit-rate falls within the (*min_target, max_target*) range.

3. AccAbove is less than AccBelow. The extra bits obtained from the video streams with their initial GOP targets greater than their *max_targets* cannot totally compensate for the insufficiency of the video streams with initial GOP targets less than their *min_targets*. Thus, the algorithm needs to take away bits from the video streams with initial GOP target in the (*min_target, max_target*) range. The taken-away bits, combined with the extra bits obtained from the videos with initial GOP target greater than their *max_target*, are distributed to the video streams that require more bits. The taken-away bits plus the extra bits are distributed as follows:

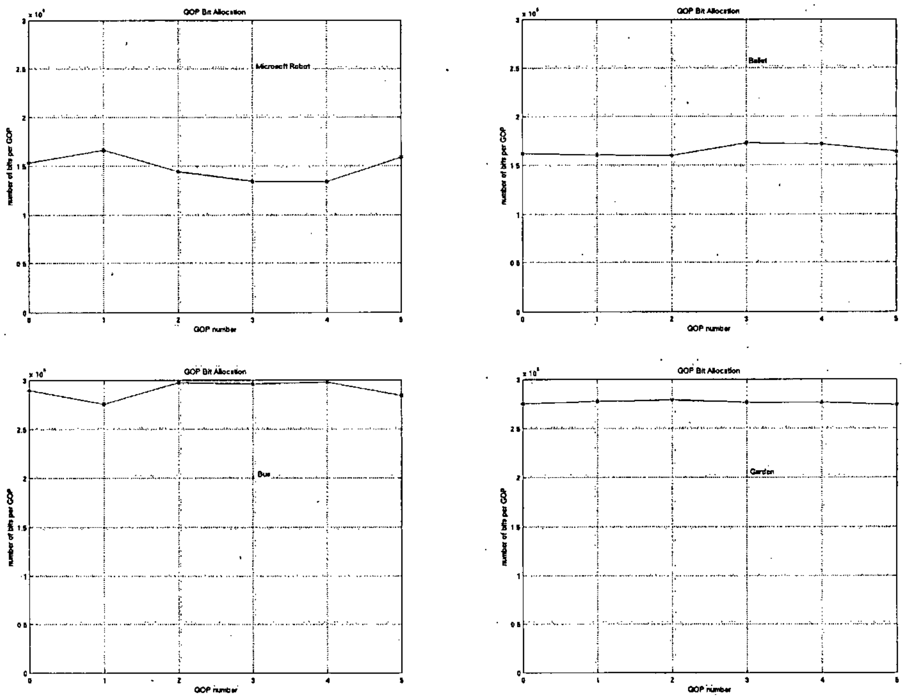
$$extra\_bits_i = \frac{GOP\_Complexity_i}{\sum_{i=1}^{N_{below}} GOP\_Complexity} * (extra\_bits + taken\_away\_bits) \qquad (3.7)$$

where, N_{below} is the number of videos that have their initial GOP target below their *min_target*.

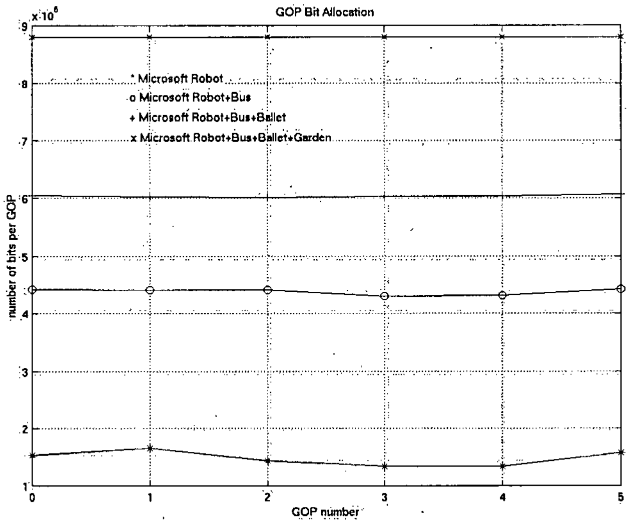
Figure 3.13 and 3.14 show the results from examining the performance of the joint bit-rate controller. The first set of video sequences consists of GARDEN, BUS, BALLET, and MICROSOFT ROBOT. Each of these four sequences is 2 seconds in length and has a 720x480 resolution. The second set of video sequences consists of four segments from the trailer of MAN IN THE IRON MASK. Each segment is 10 seconds in length and has a 720x480 resolution. All video sequences from the two sets are encoded at 30 frames/sec with a fixed quantizer scale of 6, a GOP size of 12, and a GOP pattern IBBPBBPBBPBB. Figure 3.13(a) and 3.14(a) illustrate the GOP complexity of each video stream. Figure 3.13(b) and 3.14(b) trace the GOP targets assigned to the two sets of video test sequences. Figure 3.13(c) and 3.14(c) show the sum of the GOP targets from each bit allocation decision. We observe that the GOP target assignments of each video stream closely match the stream's GOP complexities. In addition, each individual video sequence is assigned with variable GOP targets, and the sum of the assigned GOP targets or the *aggregate_GOP_target* at each instance is constant.



(a) GOP complexity of each video stream.

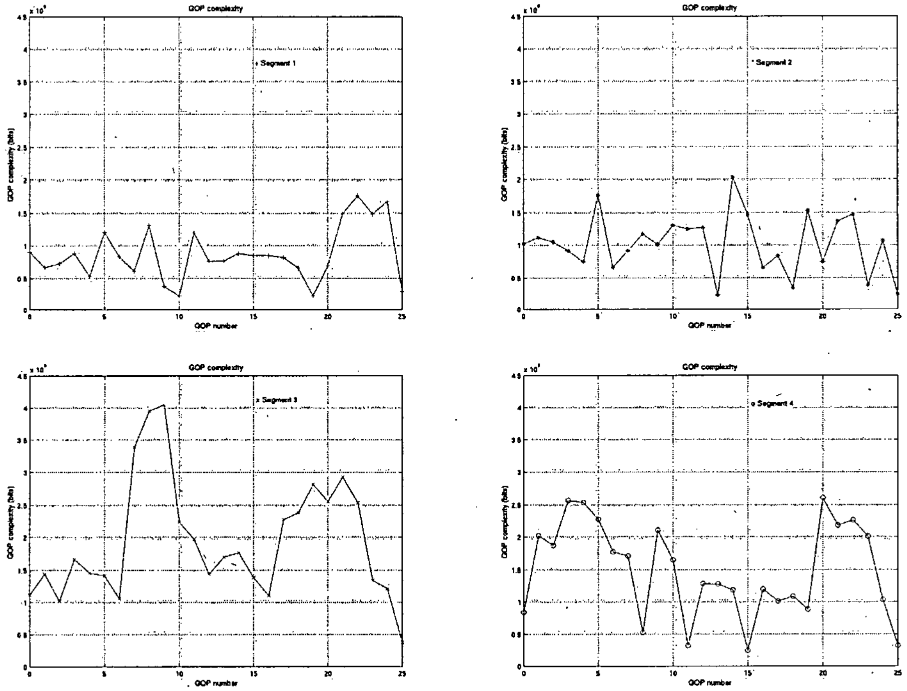


(b) GOP targets assigned to individual video segment.

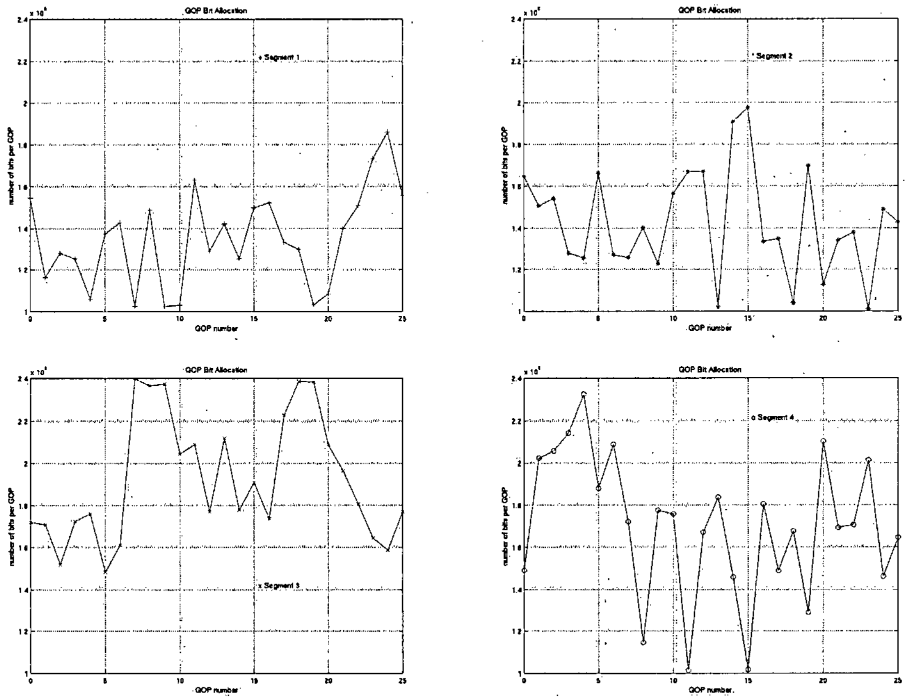


(c) Sum of GOP targets assigned to all video segments.

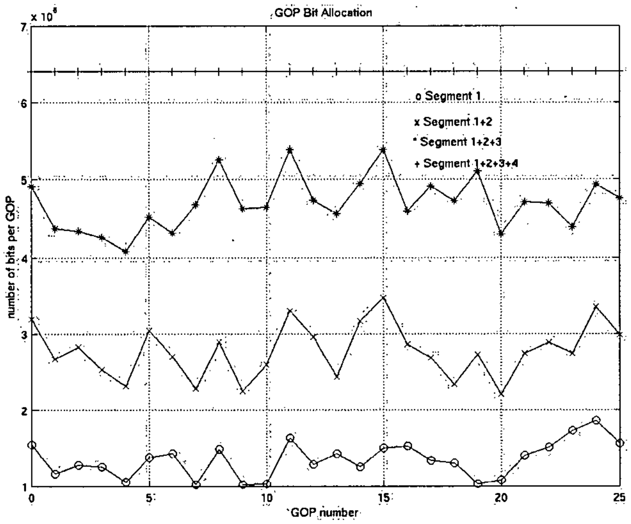
Figure 3.13: GOP bit allocation performed by the joint bit-rate controller for MICROSOFT ROBOT, BALLET, BUS, and GARDEN.



(a) GOP complexity of each video stream.



(b) GOP targets assigned to individual video segment.



(c) Sum of GOP targets assigned to all video segments.

Figure 3.14: GOP bit allocation performed by the joint bit-rate controller for SEGMENT 1, SEGMENT 2, SEGMENT 3, and SEGMENT 4 of the MAN IN THE IRON MASK trailer.

3.3.3 Parameters for Optimizing Transcoders' Performance

As shown earlier, our joint bit-rate controller uses the GOP complexities recorded in the complexity files to determine the GOP targets for each video stream. Then, this information is sent to the transcoders. Each transcoder, upon receiving its corresponding GOP target, distributes it to the pictures in the GOP. The joint bit-rate controller can help the transcoders in optimizing their performance by providing them with the picture complexities or picture targets derived from the picture complexities.

3.3.3.1 First Parameter Set: Picture Complexities

Sending picture complexities as additional information to the transcoders is beneficial. It eliminates the need for the transcoder to estimate the picture complexities from previously coded pictures. Instead, the transcoders now have accurate measurements on the complexity of each picture without performing any analysis. By making use of these received picture complexities, the transcoders can determine for each picture a picture target that reflects the picture content.

3.3.3.2 Second Parameter Set: Target Picture Bit-Rates

By using the picture complexities recorded in complexity files and the GOP target determined, the joint bit-rate controller could actually perform the picture bit allocation for the transcoders. As a result, the complexity of a transcoder can be reduced and the encoding delay in the transcoding stage can be shortened. The picture target of a picture is given as follows:

$$Picture\_Target_i = \frac{Picture\_Complexity_i}{\sum_{j=1}^{N} Picture\_Complexity_j} * GOP\_Target \qquad (3.8)$$

where, N is the number of pictures in a GOP; *Picture_Complexity* is the picture complexity of each picture recorded in the complexity file; *GOP_Target* is the GOP target determined by the joint bit-rate controller.

3.4 Transcoders

Transcoding is the final step of the joint bit-rate coding process. It is performed immediately before multiplexing and transmission of the video streams. The first half of the transcoding process partially decodes the video stream up to the stage where all DCT coefficients of macroblocks are obtained. The latter half of the transcoding process re-quantizes the DCT coefficients and puts the video stream back together. Thus, our transcoding process involves variable length decoding, inverse scanning, inverse quantization, re-quantization, forward scanning, and variable length encoding of the incoming video stream. Figure 3.15 shows the block diagram of the transcoding process.

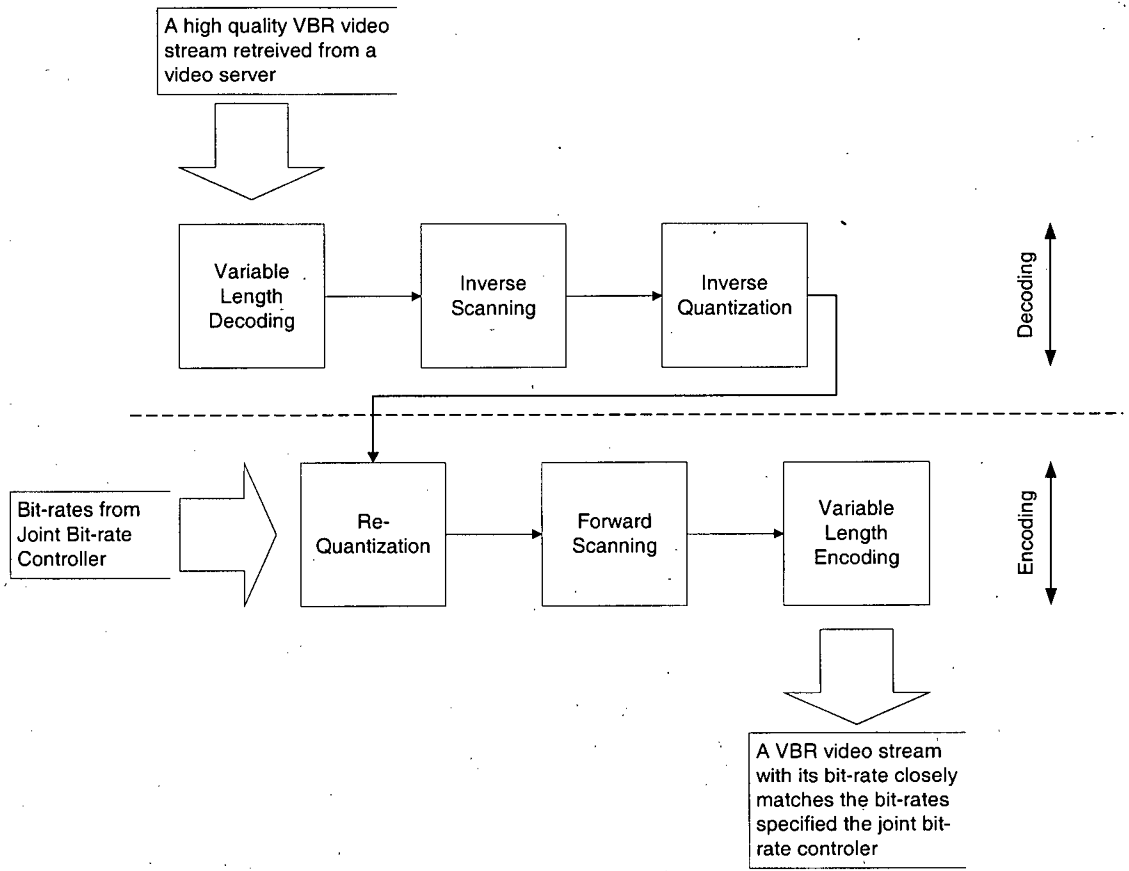

A high quality VBR video stream retreived from a video server
Variable Length Decoding
Inverse Scanning
Inverse Quantization
Decoding
Bit-rates from Joint Bit-rate Controller
Re-Quantization
Forward Scanning
Variable Length Encoding
Encoding
A VBR video stream with its bit-rate closely matches the bit-rates specified the joint bit-rate controler

Figure 3.15: Block diagram of the transcoding process.

A transcoder, essentially, consists of a cascaded decoder and encoder [20]. The complexity of a transcoder can range from the most complex, where it comprises a complete decoder and a complete encoder, to the simplest, where it is just a re-quantizer. Our transcoder implementation takes on the simplest approach. We can do so because the objective for our transcoding process is to compress the video stream from a high bit-rate to a lower bit-rate suitable for transmission. That is, no other reformatting such as resampling is involved. Since re-quantization is the sole purpose of our transcoding process, changes on performing DCT, the picture types, carrying a new set of coding decision, or a re-estimation of motion vectors is not required. Therefore, all these information, already obtained during the first stage, can be used. By reusing the set of coding decisions and the set of motion vectors, we reduce the transcoder's complexity and the processing delay.

3.4.1 The Transcoding Processing

Two functions are performed by our transcoder:

1. an initialization procedure to obtain all the necessary information to be used during transcoding and
2. a GOP processing procedure to transcode the current GOP from the initial high bit-rate to the desired target GOP bit-rate.

Figure 3.16 illustrates the dataflow diagram of the transcoding processing.

3.4.1.1 Initialization

During initialization, the transcoder

1. receives information such as N, M, and the number of frames in the video stream from the joint bit-rate controller;
2. decodes Sequence Header and all other sequence header extensions to retrieve information about the video;
3. initializes sequence rate control. The initialization of sequence rate control consists of the initial estimation of picture complexity and the initial estimation of virtual buffer fullness for each picture coding type.

3.4.1.2 Transcoding Procedure for a Group of Pictures

For every GOP, the transcoder performs the following steps:

1. it decodes the GOP Header;

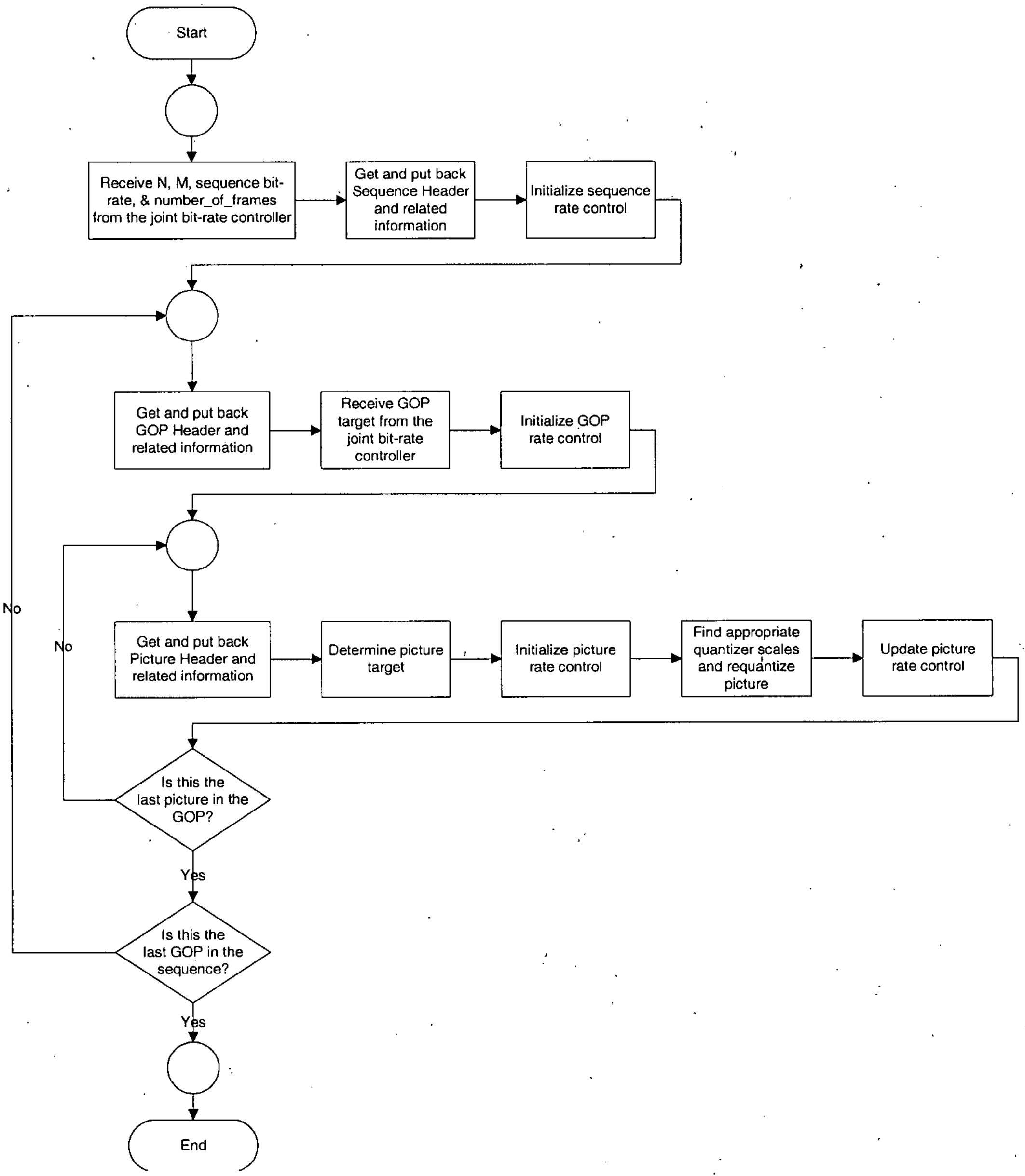

Start
Receive N, M, sequence bit-rate, & number_of_frames from the joint bit-rate controller
Get and put back Sequence Header and related information
Initialize sequence rate control
Get and put back GOP Header and related information
Receive GOP target from the joint bit-rate controller
Initialize GOP rate control
Get and put back Picture Header and related information
Determine picture target
Initialize picture rate control
Find appropriate quantizer scales and requantize picture
Update picture rate control
Is this the last picture in the GOP?
No
Yes
Is this the last GOP in the sequence?
No
Yes
End

Figure 3.16: Dataflow diagram of the transcoding process.

2. it receives from the joint bit-rate controller a GOP target for the current GOP;
3. it initializes GOP rate control;
4. for each picture in the GOP,
    a. it decodes the Picture Header and all other picture extension headers;
    b. it determines the picture target for each picture;
    c. it initializes picture rate control;
    d. it determines the appropriate quantizer scale parameters to re-quantize the picture;
    e. it updates the picture rate control parameters.

The GOP rate control initialization records the received GOP target obtained from the joint bit-rate controller. The picture target for the next picture in the GOP is defined as in the TM5 picture target procedure [4]:

$$T_i = \max\left\{\frac{R}{1+\frac{N_pX_p}{X_iK_p}+\frac{N_bX_b}{X_iK_b}}, \frac{bit\_rate}{8\times frame\_rate}\right\}$$

$$T_p = \max\left\{\frac{R}{N_p+\frac{N_bK_pX_b}{X_pK_b}}, \frac{bit\_rate}{8\times frame\_rate}\right\} \qquad (3.9)$$

$$T_b = \max\left\{\frac{R}{N_b+\frac{N_pK_bX_p}{X_bK_p}}, \frac{bit\_rate}{8\times frame\_rate}\right\}$$

where, K_p and K_b are constants defaulted to be 1.0 and 1.4 respectively; N_p and N_b are the number of P- and B- pictures remaining in the current GOP in the coder; X_i, X_p, and X_b are the estimates of the picture complexity of the next I-, P-, and B- pictures; R is the remaining number of bits assigned to the GOP. The picture complexity estimates, X_i, X_p, and X_b, are defined as follows [4]:

$$\begin{aligned} X_i &= S_i Q_i \\ X_p &= S_p Q_p \qquad (3.10) \\ X_b &= S_b Q_b \end{aligned}$$

where, S_i, S_p, and S_b are the number of bits used to code the previous I-, P-, B- picture, and Q_i, Q_p, and Q_b are the average quantizer scale parameter used to encode all macroblocks of the previous I-, P-, and B- pictures. During the initialization of picture rate control, the initial quantizer scale parameter for the picture is estimated. This quantizer scale parameter is then refined for each macroblock within the picture. The transcoder uses this refined quantizer scale parameter to re-quantize the macroblock. After the entire picture is re-quantized, the remaining number of bits, R, is updated.

3.4.2 Picture Bit-Rate Distribution

In Section 3.3.3, we introduced two sets of parameters, the picture complexities and the picture targets, which the joint bit-rate controller can send to the transcoder to help optimize picture bit distribution. In this section, we discuss the necessary changes to be made to the transcoder in order to take advantage of these two parameter sets.

3.4.2.1 Transcoding with Given Picture Complexities

The first parameter set consists of picture complexities. As discussed in Section 3.3.3.1, the joint bit-rate controller sends to the transcoder a GOP target for the current GOP and the picture complexity of the pictures in that GOP for every GOP in the video stream. The transcoder, thus, receives these two pieces of information from the joint bit-rate controller at the beginning of each GOP transcoding procedure. Step 4 of the transcoding procedure discussed in Section 3.4.1.2 becomes

4. for each picture in the GOP,
   a. it decodes the Picture Header and all other picture extension headers;
   b. it determines the picture target for each picture using the picture complexities received;
   c. it initializes picture rate control;
   d. it determines the appropriate quantizer scale parameters to re-quantize the picture;
   e. it updates the picture rate control parameters.

The picture target for each picture coding type is also determined according to Equation 3.9. However, instead of using Equation 3.10 to estimate picture complexities, the received picture complexities are used.

3.4.2.2 Transcoding with Given Picture Bits Distribution

As discussed in Section 3.3.3.2, the second set of parameters is the picture target. In this case, the transcoder also receives two pieces of information from the joint bit-rate controller at the beginning of each GOP transcoding procedure:

1. the GOP target for the current GOP;
2. a set of picture targets for the pictures within this GOP. The picture targets are computed using Equation 3.8.

Therefore, step 4 of the transcoding procedure becomes

4. for each picture in the GOP,
   a. it decodes the Picture Header and all other picture extension headers;
   b. it initializes picture rate control;
   c. it determines the appropriate quantizer scale parameters to re-quantize the picture;
   d. it updates the picture rate control parameters.

Figure 3.17 and Table 3.2 compare the picture quality of the segments from the MAN IN THE IRON MASK trailer, which were transcoded using GOP target information only, with those transcoded using GOP target as well as picture complexities. The variable bit-rate assigned to each video segment is determined by the joint bit-rate controller. Table 3.3 summarizes the bit-rates assigned to the video segments. From Table 3.2, we observe that there is an average of 0.2 dB improvement in picture quality resulting from using picture complexities.

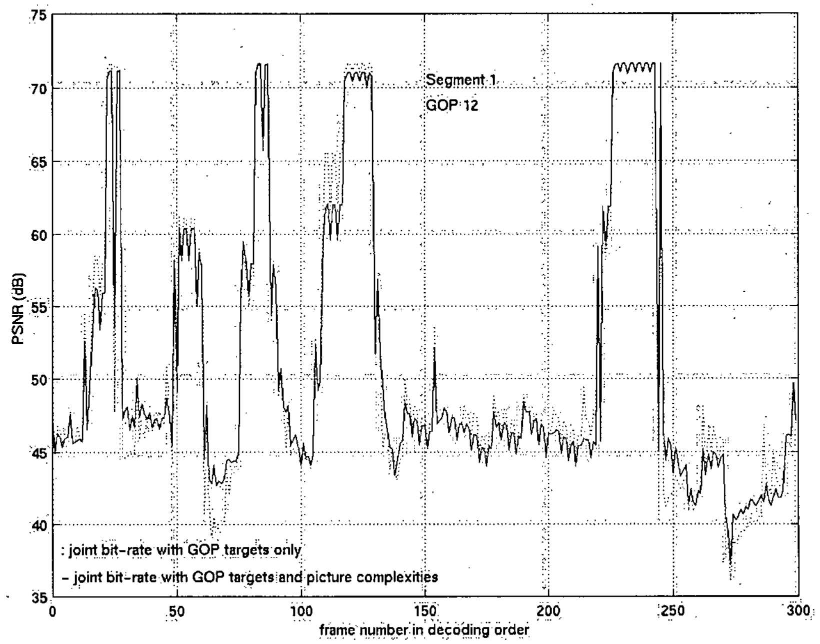
Figure 3.18 and Table 3.2 compare the picture quality of the MAN IN THE IRON MASK trailer segments transcoded using GOP targets only with those transcoded using both GOP targets and picture targets. The same bit-rates shown in Table 3.3 are assigned to video segments. The results from Table 3.2 show that the addition of picture targets gives an average 0.9 dB improvement in picture quality.

Coding Algorithm	Average PSNR	Std. Dev. PSNR	Max. PSNR	Min. PSNR
Segment 1				
Joint using GOP targets only	50.97	9.90	71.60	36.10
Joint using GOP targets and picture complexities	50.94	9.54	71.60	37.20
Joint using GOP and picture targets	51.79	9.37	71.70	40.70
Segment 2				
Joint using GOP targets only	50.88	11.00	71.60	37.80
Joint using GOP targets and picture complexities	51.35	11.00	71.60	39.20
Joint using GOP and picture targets	52.17	10.30	71.70	39.40
Segment 3				
Joint using GOP targets only	43.16	5.76	54.10	33.80
Joint using GOP targets and picture complexities	43.46	5.59	64.40	32.80
Joint using GOP and picture targets	44.24	5.68	65.40	35.50
Segment 4				
Joint using GOP targets only	74.23	9.54	71.60	36.90
Joint using GOP targets and picture complexities	47.29	9.59	71.60	36.90
Joint using GOP and picture targets	47.62	9.31	71.60	37.80

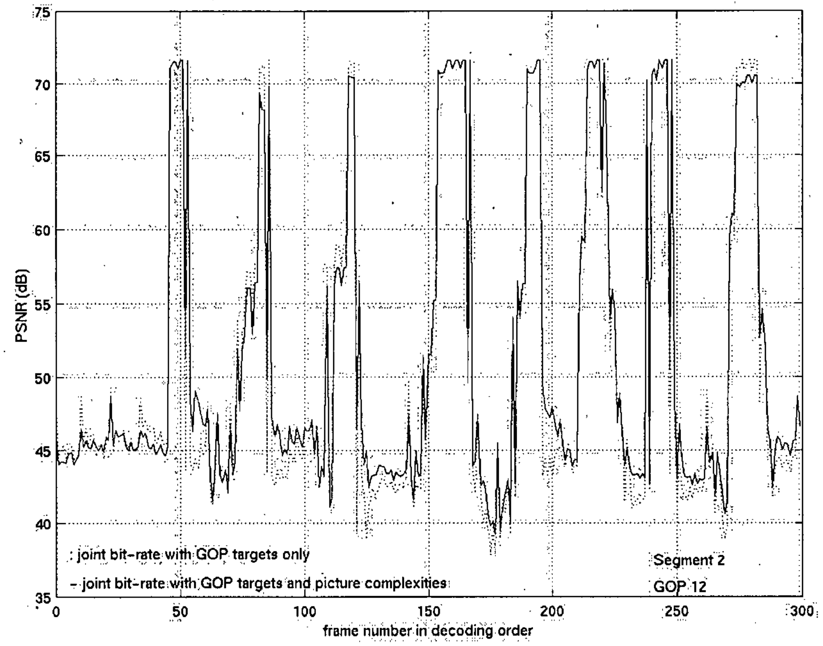
Table 3.2: Comparison of PSNR values for video streams encoded with joint bit-rate coding using GOP targets only, with joint bit-rate coding using GOP targets and picture complexities, and with joint bit-rate coding using GOP and picture targets.

	Average Bit-Rate (Mbps)	Maximum Bit-Rate (Mbps)	Minimum Bit-rate (Mbps)
Segment 1	3.37	4.65	2.56
Segment 2	3.56	4.94	2.53
Segment 3	4.80	5.99	3.71
Segment 4	4.26	5.81	2.53

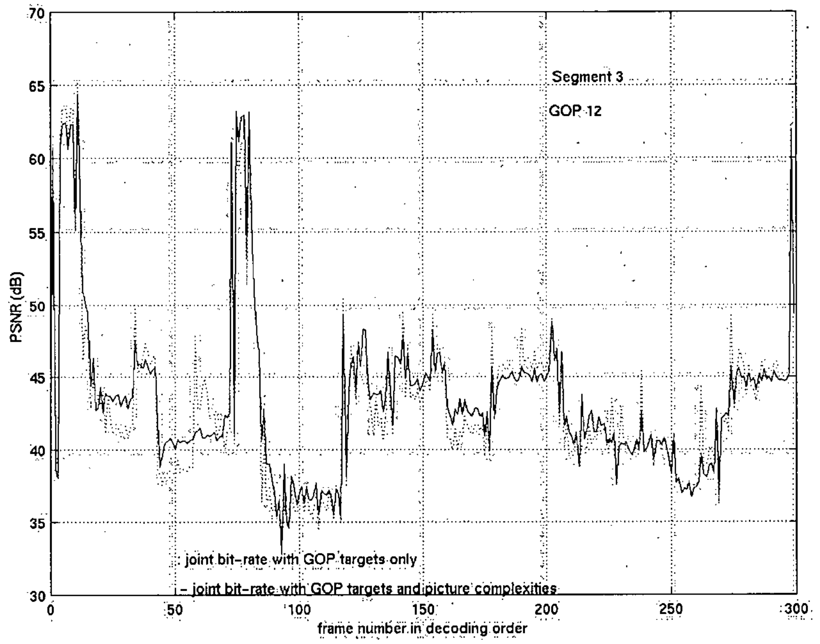
Table 3.3: Bit-rates of the four MAN IN THE IRON MASK video segments assigned by the joint bit-rate controller.



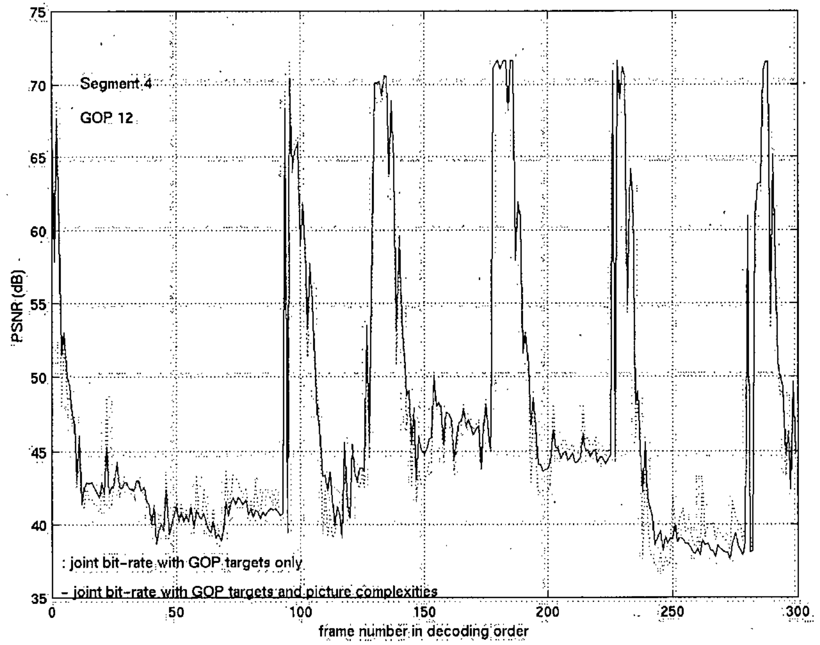
(a) Segment 1



(b) Segment 2

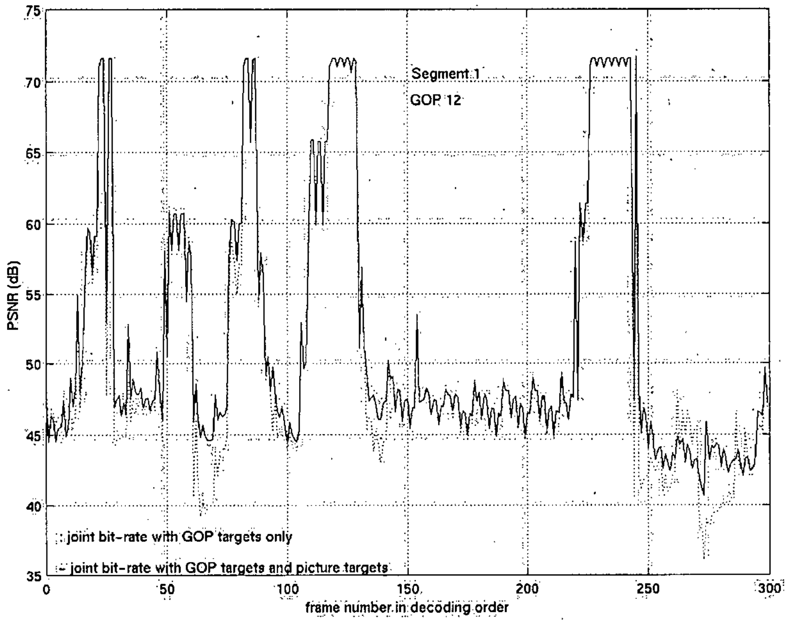


(c) Segment 3

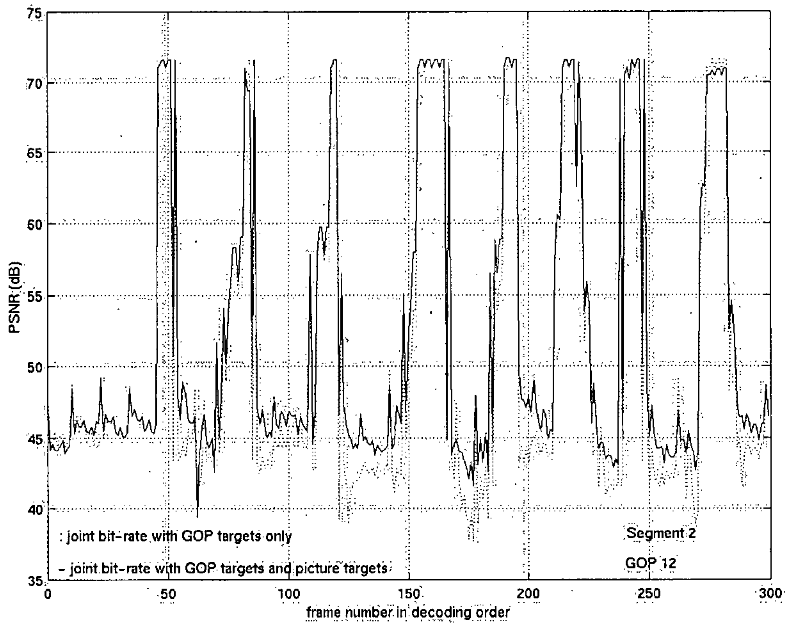


(d) Segment 4

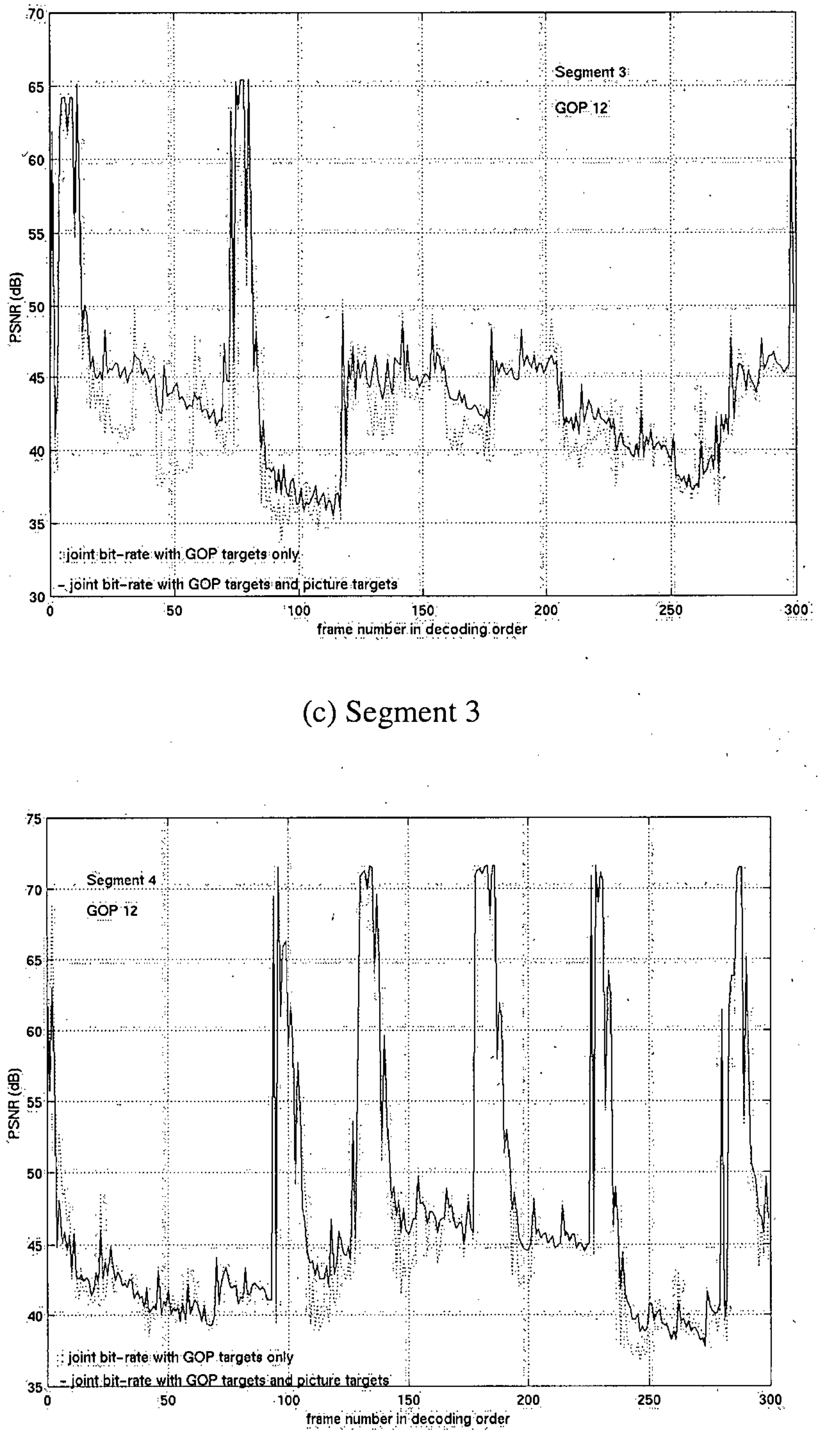
Figure 3.17: The picture quality of the MAN IN THE IRON MASK trailer segments transcoded with GOP targets only and the picture quality of the same segments transcoded with GOP targets and picture complexities.



(a) Segment 1



(b) Segment 2



(c) Segment 3

(d) Segment 4

Figure 3.18: The picture quality of MAN IN THE IRON MASK trailer segments transcoded with GOP targets only and the picture quality of the same segments transcoded with GOP and picture targets.

3.5 Summary

In this chapter, we presented our two-stage joint bit-rate coding system that is intended for the video coding of multiple video sources for transmission in a constant bit-rate medium. The system focuses on providing consistent picture quality to each video stream present in the system and on allocating to each video stream an appropriate portion of the given channel bandwidth, which is dependent on the picture complexities.

In order to achieve the goals of our system, we use a two-stage coding approach. The first stage analyzes the activity level of the pictures in each video stream and records the data in complexity files. It also facilitates the processing of the second stage by performing motion compensation predictions on the video sources. By using the results from the analyses performed during the first stage and the knowledge of the available channel bandwidth, the joint bit-rate controller determines the number of bits needed to encode a GOP of each video stream and send these GOP targets to a set of transcoders. The transcoders carry out the bit allocation decisions by re-quantizing their corresponding video streams. As an enhancement to the system, the joint bit-rate controller also sends a set of picture complexities or a set of picture targets to each transcoder, facilitating the picture bit allocation process during the transcoding stage.

In the simulation results to be discussed in the next chapter, we will show that our two-stage joint bit-rate coding system significantly reduces the fluctuations in picture quality of all video streams. We will also show that the bit allocation decisions performed by our system reflect the complexities of the video streams.

Chapter 4 Simulation Results and Discussions

In the previous chapter, we presented our two-stage joint bit-rate coding system for coding multiple video programs simultaneously. The system is designed to provide more efficient usage of the available bandwidth. It is also designed to offer each video stream more consistent picture quality than that obtained by coding each video stream individually at a constant bit-rate. By using the two-stage approach, the system also reduces the coding delay introduced immediately before the multiplexing and transmission of the video streams. To test the performance of our system, we compare the bandwidth consumption of each individual video stream encoded using our coding system with the bandwidth consumption of the same video stream encoded using the Test Model 5 [4] algorithm. To illustrate the picture quality consistency provided by our system, we compare the standard deviations of the picture quality (PSNR) of the reconstructed images obtained from our coding system with the standard deviation of the picture quality of those images obtained by independent CBR coding. We also compare the CPU time used to encode a video stream using our system and that using the TM5 algorithm [4].

In this chapter, we present the simulation results of the tests. In Section 4.1 we describe the setup of the tests. In Section 4.2, we show the bit-rates as well as the PSNR standard deviations of the test sequences obtained by using our system and by independent CBR coding. In Section 4.3, we present the results from timing analysis.

4.1 Setup

Two sets of simulations were carried out using our joint bit-rate coding system. The first set involves six video sequences at a total bit-rate of 20 Mbps. The video sequences are segments extracted from the trailer of THE MAN IN THE IRON MASK. The second set involves five video sequences at a total bit-rate of 18 Mbps. The video sequences comprise extremely complex scenes and less complex scenes from various video clips. Each video sequence is 10 seconds in length and has a spatial resolution of 720x480. This resolution is approximately double the ones used in present broadcast systems. For example, DirectTV uses a spatial resolution of 545x480 for satellite transmission; GI uses 368x480 for cable transmission; TCI uses 352x480 for cable transmission. Therefore, the bit-rates of the resulting video streams are expected to be higher than those encountered in present systems. All video sequences were interlaced, encoded at a frame rate of 30 frames/s with a color sampling ratio of 4:2:0. The GOP pattern used in each sequence is IBBPBBPBBPBB. With the option of providing additional information, the joint bit-rate controller is set to send both GOP targets and picture targets to the set of transcoders.

4.2 Simulation Results

The MAN IN THE IRON MASK trailer is composed of various scenes from the movie. Between scenes, black frames have been inserted to signal the change of scenes. These black frames, when encoded, have extremely high PSNR values (above 55 dB). We have chosen to ignore these frames in the discussion of picture quality fluctuation since their inclusions in the analysis would give a bias to the results.

Table 4.1(a) shows the bit-rates for the six THE MAN IN THE IRON MASK sequences encoded using joint bit-rate coding. Table 4.1(b) shows the average GOP complexity of the same six video sequences. Table 4.2(a) shows the bit-rates for the five video sequences from the second test set encoded using our joint bit-rate coding system. Table 4.2(b) shows the average GOP complexity of the same five sequences. It is evident that with joint bit-rate coding, each video sequence was encoded with different number of bits depending upon its complexity. For example, from THE MAN IN THE IRON MASK test sequences, SEGMENT 3 was encoded using 30% more bits than SEGMENT 2. The square-rooted complexity ratio of SEGMENT 3 to SEQUENCE 2, $\frac{\sqrt{Average\_Complexity_{Segment3}}}{\sqrt{Average\_Complexity_{Segment2}}}$, is 1.38. That is, the square-rooted complexity of SEGMENT 3 is 38% higher than that of SEGMENT 2. Using the second set of video sequences as another example, the average bit-rate ratio of SEQUENCE 1 to SEQUENCE 2 is 1.02, and the square-rooted complexity ratio of SEQUENCE 1 to SEQUENCE 2 is 1.03. Therefore, the bit assignments and the relative complexities of a video sequence are closely related. Figure 4.1 and Figure 4.2 show the GOP complexities and bit-rates assigned by the joint bit-rate controller for each of the six MAN IN THE IRON MASK sequences and each of the five video sequences from the second test set respectively. There is a high degree of resemblance between the GOP complexity plot and the bit-rate plot of each video sequence. It is evident that the bit-rate of each video sequence varies in time and in accordance with the complexity of the GOP in that instance.

	Average (Mbps)	Standard Deviation (Mbps)	Maximum (Mbps)	Minimum (Mbps)
SEGMENT 1	2.90	0.33	3.66	2.52
SEGMENT 2	3.03	0.42	3.86	2.55
SEGMENT 3	3.95	0.58	5.17	3.10
SEGMENT 4	3.56	0.72	4.93	2.55
SEGMENT 5	3.08	0.53	3.98	2.50
SEGMENT 6	3.47	0.36	3.99	2.52

Table 4.1(a): Bit-rates of the MAN IN THE IRON MASK video sequences encoded using the joint bit-rate coding system.

	Average Complexity
SEGMENT 1	870560
SEGMENT 2	1022600
SEGMENT 3	1945100
SEGMENT 4	1491000
SEGMENT 5	1101000
SEGMENT 6	1671900

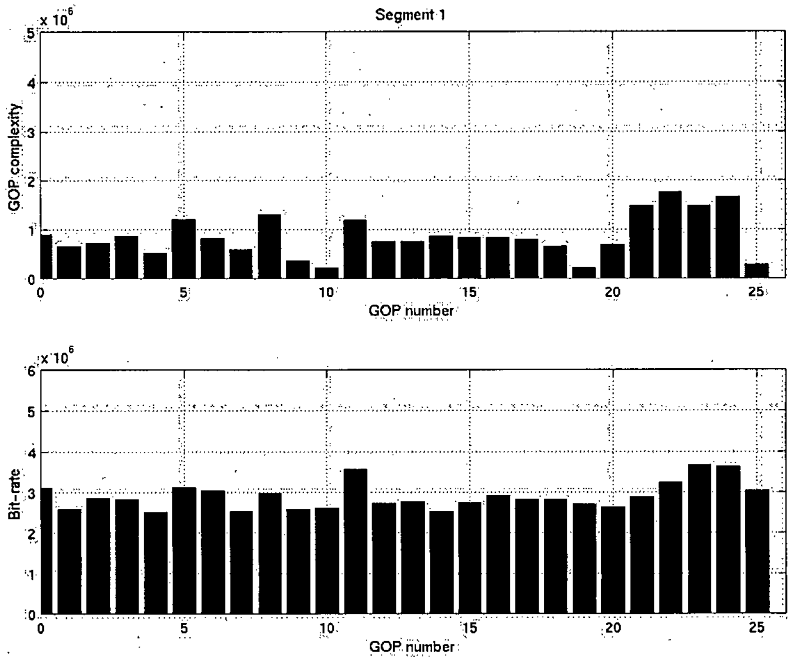
Table 4.1(b): Average GOP complexity of the MAN IN THE IRON MASK video sequences.

	Average (Mbps)	Standard Deviation (Mbps)	Maximum (Mbps)	Minimum (Mbps)
SEQUENCE 1	3.79	1.71	8.02	1.88
SEQUENCE 2	3.73	1.52	7.38	1.96
SEQUENCE 3	3.52	1.58	7.27	2.06
SEQUENCE 4	3.23	1.61	7.78	1.94
SEQUENCE 5	3.73	1.46	7.76	2.39

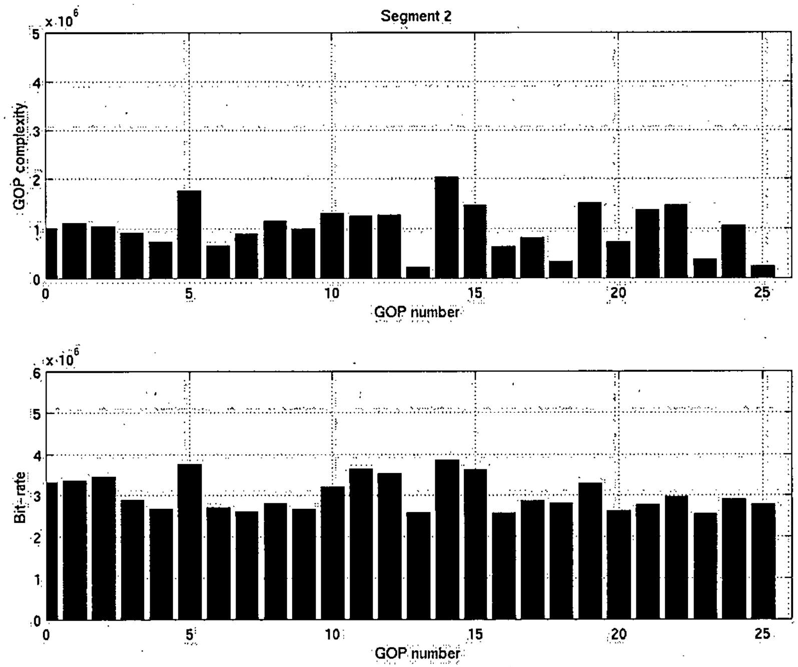
Table 4.2(a): Bit-rates of the video sequences from the second test set encoded using the joint bit-rate coding system.

	Average Complexity
SEQUENCE 1	2974767
SEQUENCE 2	2778334
SEQUENCE 3	2720989
SEQUENCE 4	2290514
SEQUENCE 5	2818206

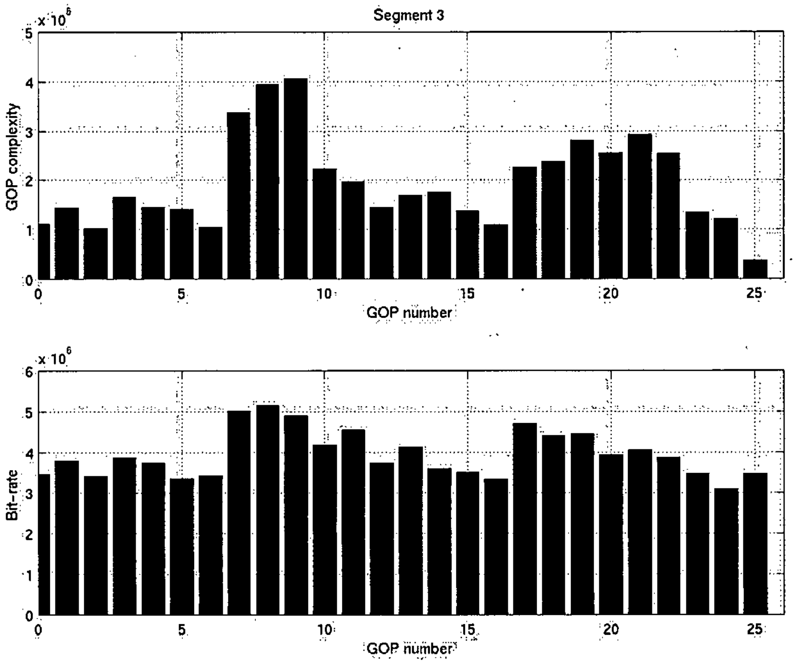
Table 4.2(b): Average GOP complexity of the video sequences from the second test set.



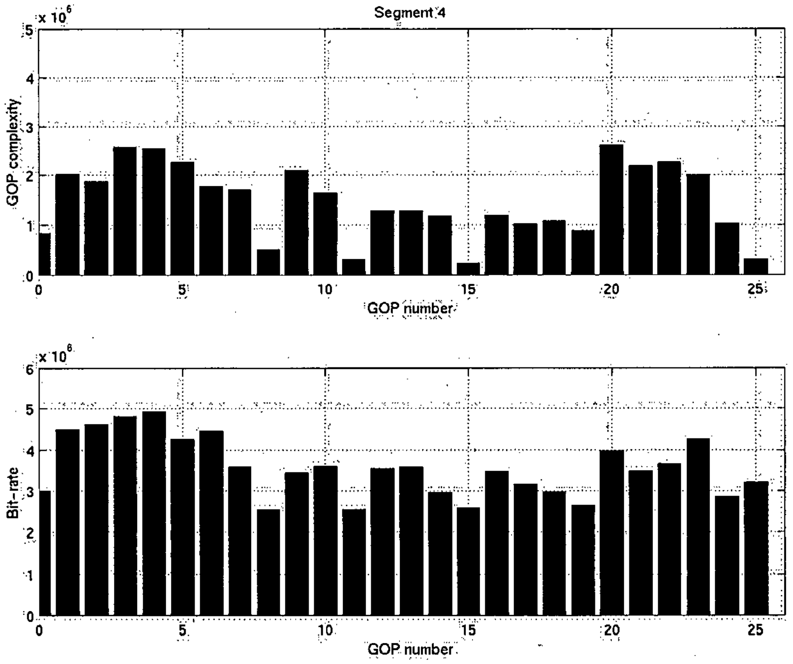
(a) SEGMENT 1



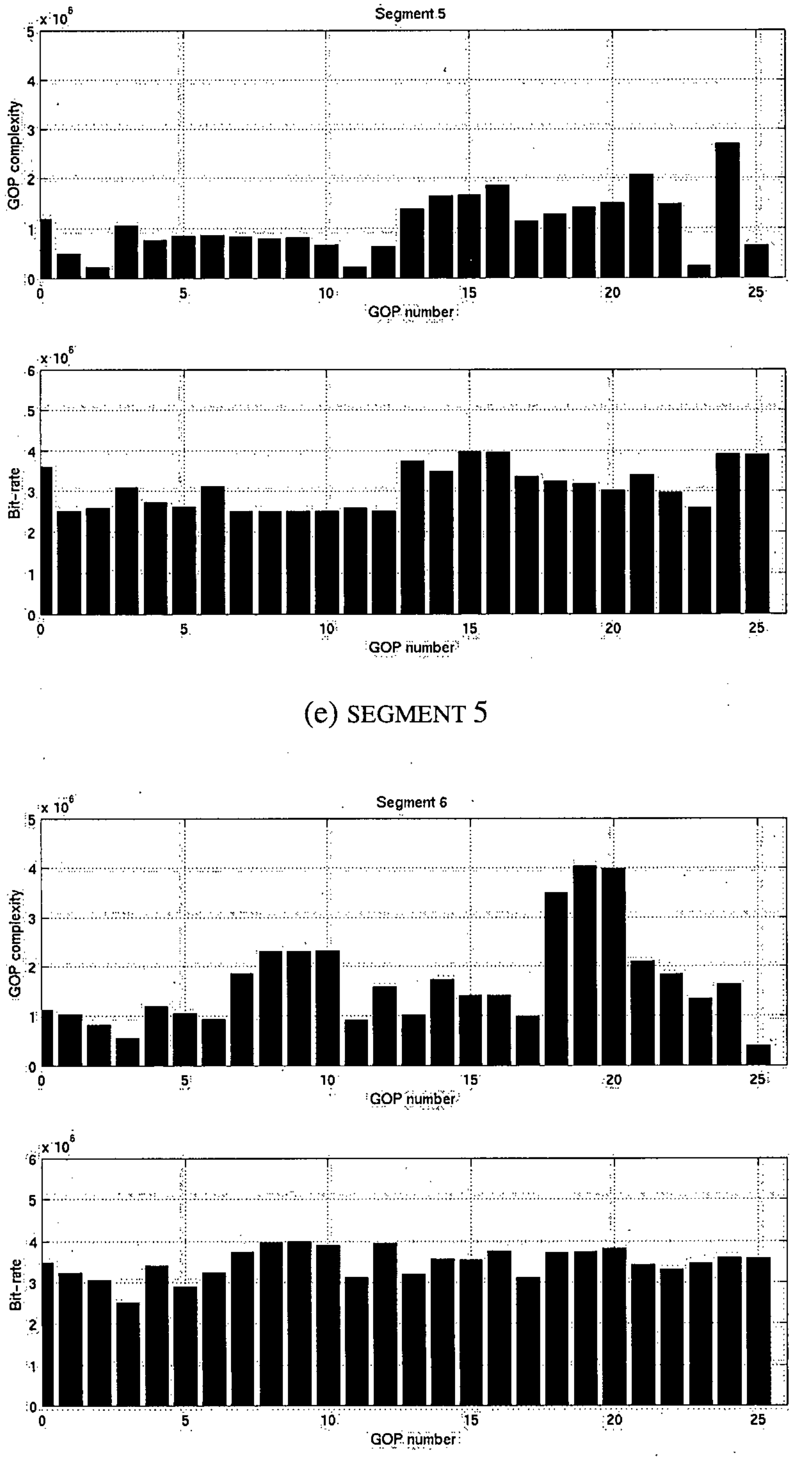
(b) SEGMENT 2



(c) SEGMENT 3



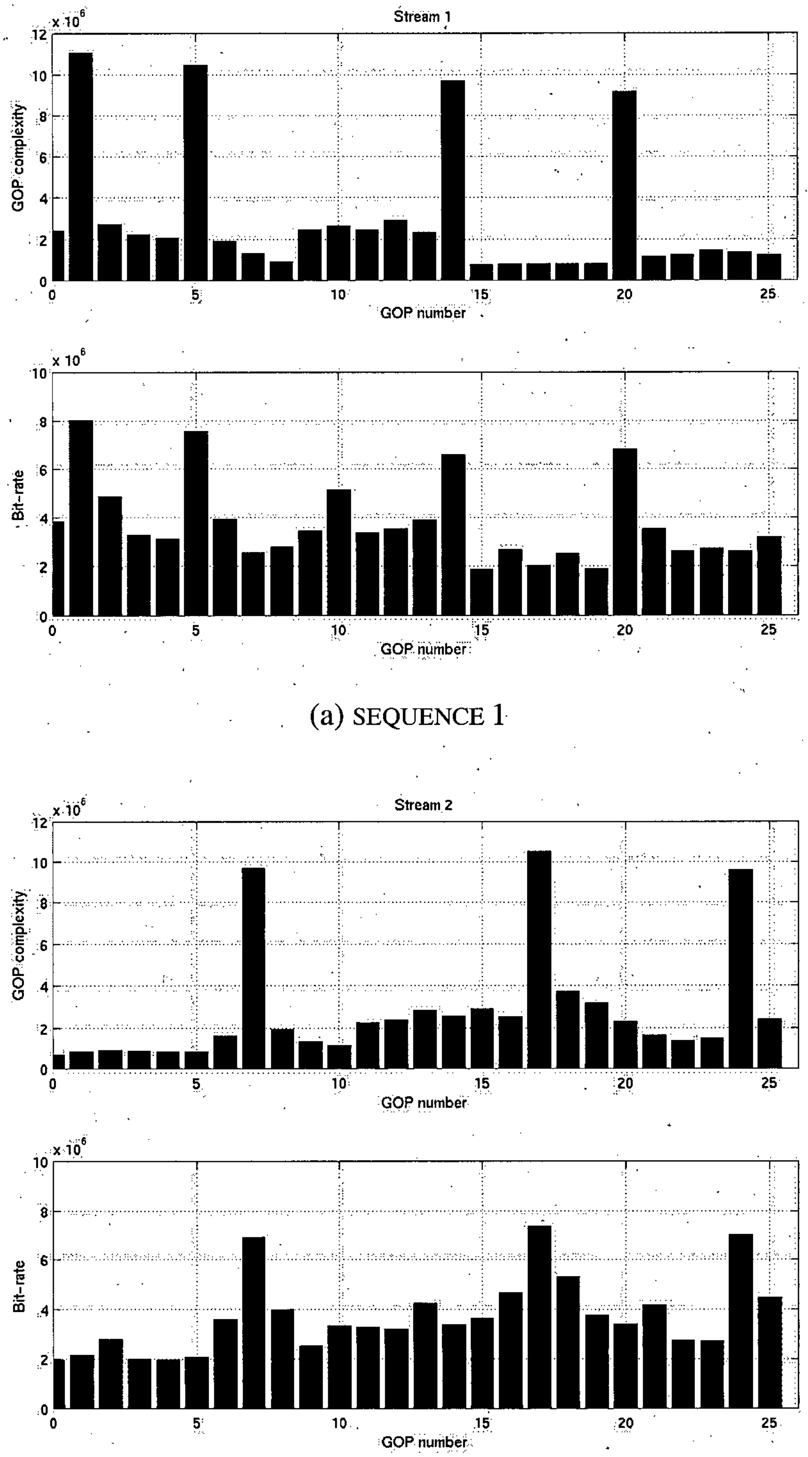
(d) SEGMENT 4



(e) SEGMENT 5

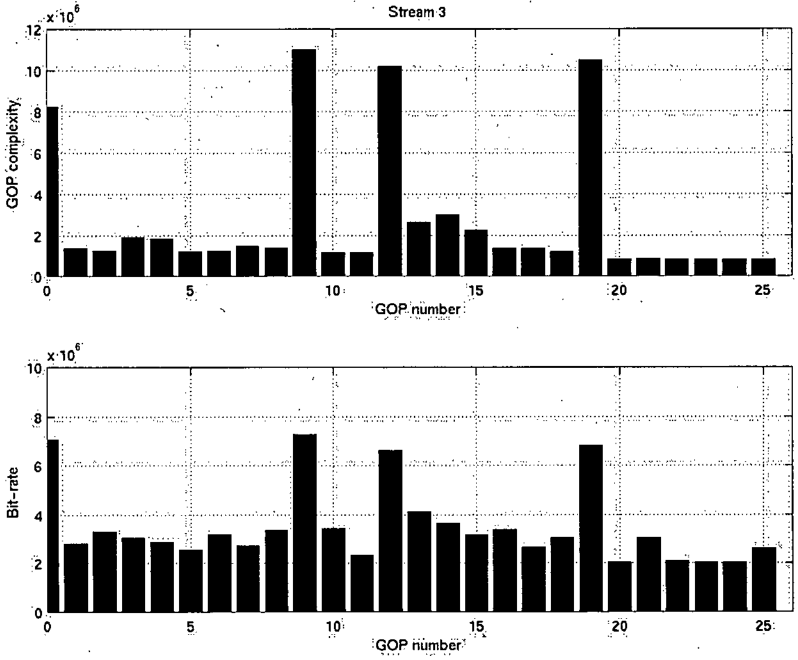
(f) SEGMENT 6

Figure 4.1: GOP complexities and bit-rates of the MAN IN THE IRON MASK video sequences.

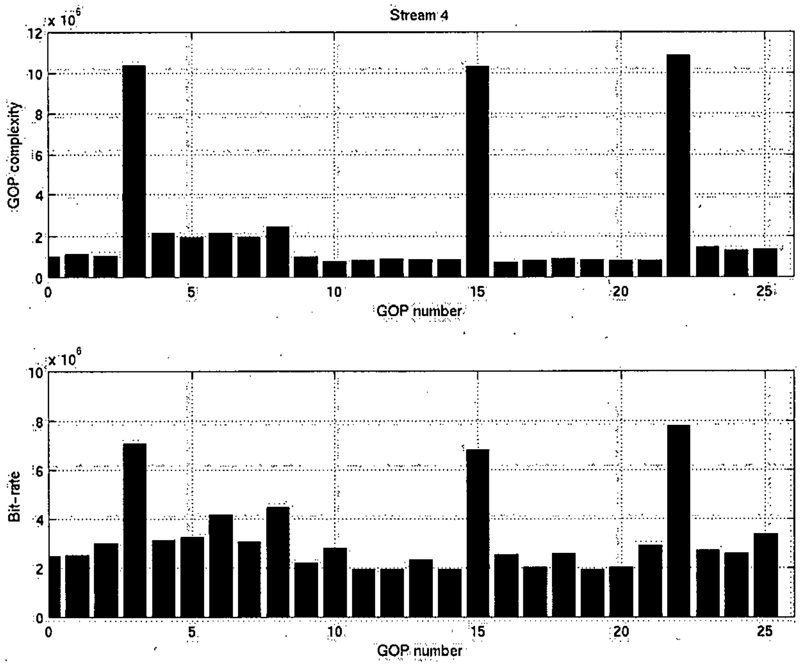


(a) SEQUENCE 1

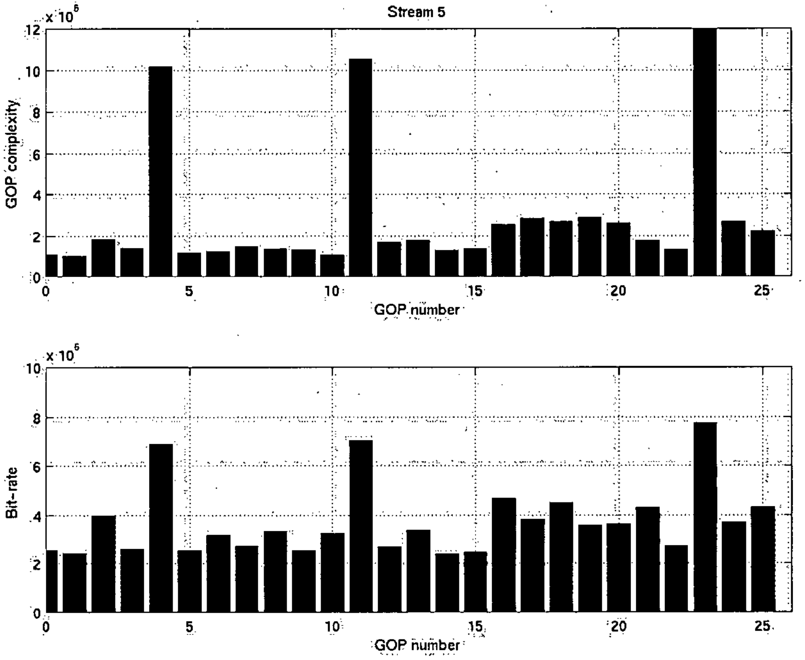
(b) SEQUENCE 2



(c) SEQUENCE 3



(d) SEQUENCE 4



(e) SEQUENCE 5

Figure 4.2: GOP complexities and bit-rates of the second set of video sequences.

Almost all video sequences have periods of highly complex scenes as well as periods of less complex scenes. If independent CBR coding were used, the bit-rate of each video stream had to be set to a high enough value to guarantee that the picture quality of the video stream during the most active segment be similar to the picture quality of the same segment obtained using our system. Using the six video sequences from THE MAN IN THE IRON MASK as an example, if the video streams were to be encoded using CBR coding, the bit-rates of the six video streams would have to be set to 3.66 Mbps, 3.86 Mbps, 5.17 Mbps, 4.93 Mbps, 3.98 Mbps and 3.99 Mbps. However, since CBR coding directly encodes the video streams while our joint bit-rate coding system re-quantizes the video streams, a slightly lower bit-rate could be used for the CBR coding of each video stream. For the six video sequences, the constant bit-rates that give the most active segment of the video streams picture quality similar to the picture quality obtained using our joint bit-rate coding system are 3.66 Mbps, 3.86 Mbps, 4.80 Mbps, 4.70 Mbps, 3.80 Mbps and 3.70 Mbps for SEGMENT 1, SEGMENT 2, SEGMENT 3, SEGMENT 4, SEGMENT 5, and SEGMENT 6 respectively. Therefore, the 20 Mbps channel would not be able to accommodate all six CBR video streams. Instead, only 4.9 video streams could be fitted into the 20 Mbps channel.

For the second set of video sequences, the constant bit-rates that give the most active segment of the video streams picture quality similar to the picture quality obtained using our system are 6.90 Mbps, 6.64 Mbps, 7.20 Mbps, 7.20 Mbps, and 6.50 Mbps for SEQUENCE 1, SEQUENCE 2, SEQUENCE 3, SEQUENCE 4, and SEQUENCE 5 respectively. The 18 Mbps channel could not accommodate all five CBR video streams. Instead, only 2.6 CBR video streams could be transmitted simultaneously down the 18 Mbps channel.

Table 4.3 and Table 4.4 summarize the standard deviations of the PSNR values obtained using our joint bit-rate coding system as well as those from CBR coding. It should be noted that for the MAN IN THE IRON MASK video sequences, the PSNR values of the black frames inserted in between scenes are not included in this analysis. The lower PSNR standard deviations from joint bit-rate coding show that our joint bit-rate coding system significantly reduces the fluctuation in picture quality of the resulting video streams. For the first set of test sequences, an average 15% reduction in picture quality fluctuation is achieved by our system. For the second set of test sequences, our system lowers the picture quality variations by 21%.

Coding Method	Standard Deviation PSNR
SEGMENT 1	
Joint bit-rate using GOP complexities and picture targets	2.68
TM5 CBR @ 3.66 Mbps	3.28
SEGMENT 2	
Joint bit-rate using GOP complexities and picture targets	2.61
TM5 CBR @ 3.86 Mbps	3.37
SEGMENT 3	
Joint bit-rate using GOP complexities and picture targets	3.47
TM5 CBR @ 4.80 Mbps	4.08
SEGMENT 4	
Joint bit-rate using GOP complexities and picture targets	3.65
TM5 CBR @ 4.70 Mbps	3.93
SEGMENT 5	
Joint bit-rate using GOP complexities and picture targets	2.81
TM5 CBR @ 3.80 Mbps	3.57
SEGMENT 6	
Joint bit-rate using GOP complexities and picture targets	4.49
TM5 CBR @ 3.70 Mbps	4.73

Table 4.3: PSNR standard deviations for MAN IN THE IRON MASK video sequences encoded using our two-stage joint bit-rate coding system and encoded independently using the TM5 method.

Coding Method	Standard Deviation PSNR
SEQUENCE 1	
Joint bit-rate using GOP complexities and picture targets	5.17
TM5 CBR @ 6.90 Mbps	6.92
SEQUENCE 2	
Joint bit-rate using GOP complexities and picture targets	4.93
TM5 CBR @ 6.64 Mbps	6.16
SEQUENCE 3	
Joint bit-rate using GOP complexities and picture targets	5.57
TM5 CBR @ 7.20 Mbps	6.92
SEQUENCE 4	
Joint bit-rate using GOP complexities and picture targets	5.18
TM5 CBR @ 7.20 Mbps	6.37
SEQUENCE 5	
Joint bit-rate using GOP complexities and picture targets	4.66
TM5 CBR @ 6.50 Mbps	6.10

Table 4.4: PSNR standard deviations for the second set of video sequences encoded using our two-stage joint bit-rate coding system and encoded independently using the TM5 method.

4.3 Timing Analysis

Our joint bit-rate control system reduces the coding delay experienced before the video streams are multiplexed and transmitted. Delay reduction is achieved by our system because motion compensation prediction was performed ahead of time. Therefore, instead of encoding a video stream completely, our system only needs to transcode the video streams to ones that have the bit-rates specified by the joint bit-rate controller. To illustrate the performance of our system in reducing coding delay, we compare the CPU time used by our transcoder in transcoding a video sequence with the CPU time used by a TM5 encoder in encoding the same video sequence. The TM5 encoder employs an exhaustive integer vector block matching algorithm for motion compensation prediction. The search window for the motion vectors of P- pictures is set to be (11,11). The two sets of search windows for both the forward and the backward motion vectors of B-pictures are {(7,7), (3,3)} and {(3,3), (7,7)}. Since we would like to emphasize the benefits in performing motion compensation predictions ahead of time, only the time used by our transcoder in performing variable length decoding, re-quantization, and variable length encoding on all macroblocks and the time used by the TM5 encoder in performing motion compensation prediction, discrete cosine transform, and variable length encoding on all macroblocks are recorded. Some I/O operations are involved in variable length coding. However, since only a few bytes are read or written in each I/O operation, the CPU time used in performing such I/O operations is assumed negligible.

Both sets of video sequences were analyzed using a Sun™ Ultra Sparc™ workstation with one Sparc™ floating point processor at 167 MHz, 128 Megabytes of RAM, and running under Solaris™ 2.5. Table 4.5 and Table 4.6 show the CPU time recorded in transcoding and encoding the two sets of video sequences that are 10 seconds in length each.

Coding	SEGMENT 1	SEGMENT 2	SEGMENT 3	SEGMENT 4	SEGMENT 5	SEGMENT 6
Joint	210.660	236.470	333.910	282.780	247.340	318.580
CBR	1865.740	1921.410	2101.440	2010.230	1890.990	2059.210
$\frac{\text{Joint}}{\text{CBR}}$	0.113	0.123	0.159	0.141	0.131	0.155

Table 4.5: CPU time used in transcoding and encoding the MAN IN THE IRON MASK video sequences.

Coding	SEQUENCE 1 (sec)	SEQUENCE 2 (sec)	SEQUENCE 3 (sec)	SEQUENCE 4 (sec)	SEQUENCE 5 (sec)
Joint	384.350	368.020	372.050	356.420	396.710
CBR	2393.262	2120.865	2176.417	2284.886	1974.297
$\frac{\text{Joint}}{\text{CBR}}$	0.161	0.174	0.171	0.156	0.201

Table 4.6: CPU time used in transcoding and encoding the second set of video sequences.

From the results shown in Table 4.5 and Table 4.6, the transcoding performed by our joint bit-rate coding system provides a huge improvement in shortening the coding delay. For the first set of test sequences, the time required for transcoding the video streams, on average, is about 13.7% of the time used in encoding the video streams completely. For the second set of test sequences, our system speeds up the coding process by 82.7%.

4.4 Summary

The performance of our two-stage joint bit-rate coding system is presented in this chapter. It is shown that the bit allocation decisions performed by the joint bit-rate controller reflect the complexities of the video streams. More video streams can be supported in a channel if joint bit-rate coding is used instead of CBR coding. For our first set of test sequences, all 6 video streams encoded using our joint bit-rate coding system are transmitted in the 20 Mbps channel while only 4.9 video streams encoded using CBR coding are transmitted in the same channel. For the second set of test sequences, 5 instead 2.6 video streams are transmitted in the 18 Mbps when our system is used. The joint bit-rate coding system is also able to reduce the picture quality variation in the video streams. An average of 15% and 21% reduction in picture quality fluctuation are achieved for the first and the second set of test sequences respectively. Simulation results have also shown that by transcoding instead of real-time encoding the video streams, our system saves about 80% of the coding time.

Chapter 5 Summary and Future Work

5.1 Summary

With the high bandwidth that is available in digital broadcasting, it is more efficient and cost-effective to multiplex several video sources together and transmit the multiplexed video stream via the fixed capacity medium. The two challenges in broadcasting multiple sources are using the available bandwidth efficiently and maintaining consistent picture quality in each of the video streams multiplexed. Currently, broadcasters either assign a fixed portion of the available channel bandwidth to each video stream or statistically multiplex the video streams for transmission. As discussed in Chapter 2, CBR coding suffers from significant fluctuations in picture quality. Since statistical multiplexing is subject to packet loss, the channel bandwidth is not efficiently used. The goal of this thesis is to develop a multiple-source video coding system that can reduce the variations in picture quality and make efficient use of the channel bandwidth.

We achieve our goals by developing a two-stage joint bit-rate coding system for simultaneous coding of multiple sources. This system can be easily implemented for commercial use in digital video broadcast applications. The system uses a two-stage approach. During the first stage, the video sources are encoded with very high picture quality and the complexities of the video streams are recorded for later use. Knowing the complexities of the video streams, the system determines the necessary number of bits needed to encode each video stream.. A set of transcoders is implemented in the system to execute the bit allocation decisions. This two-stage encoding system is intended for video that was archived and not for video programs to be transmitted live.

When comparing to present broadcast systems, for the same picture quality our system greatly increases the number of video streams transmitted in each channel. Simulation results have shown that our two-stage joint bit-rate coding system increases the number of video streams supported from 4.9 to 6 in a 20 Mbps channel and from 2.6 to 5 in an 18 Mbps channel. The results show a 22% and a 92% improvement. Such improvements are very significant since a large number of the transponders can be freed up to carry real-time video programs or to provide other communication services. By switching from tape storage to video server technology, playback systems are eliminated since video streams can be directly accessed via the video server. Presently, an encoder is needed for each video stream to be transmitted. However, since the first-stage video encoding of our system is an off-line process, fewer complete encoders are required for our system. The simpler structure of a transcoder makes the manufacturing of the transcoder hardware much easier than the manufacturing of the encoder hardware. Both of these properties translate to a lower cost at the headend. In addition to the gain in bandwidth and the reduction in cost, simulation results have also shown that our system reduces picture quality fluctuation by 15% ~ 21% and that it speeds up the coding process by 82 ~ 87%.

5.2 Future Work

Our two-stage joint bit-rate coding system is developed for applications that involve the broadcasting of pre-recorded video. We suggest the following modifications, which will increase bandwidth saving.

In our implementation, video sources are encoded into MPEG-2 video streams during the first stage. The picture complexities and GOP complexities of video sources are also recorded. Compressed video streams along with their complexity files are stored in video servers for reducing the cost at the headend. However, the re-quantization of the pre-compressed streams reduces the quality of the transmitted video. They only way to avoid degradation of quality or improve bandwidth allocation is to eliminate the re-quantization process. In this case, we can consider the following two different approaches:

1. During the first stage, we only store the complexity information. The transcoders are replaced by "complete" encoders, and VTR's are used for playing the original materials.
2. During the first stage, we store the complexity information as well as motion estimation and motion compensation decisions. In this case, the transcoders have to be modified to accept this information so that no motion estimation is needed at the second stage.

Both of the above implementations yield the same picture quality or bandwidth savings. However, feasibility studies are needed to determine which approach is more cost-effective.

Chapter 6 Bibliography

[1] ISO/IEC 13818, "Information Technology – Generic coding of moving pictures and associated audio information." November 1994

[2] ISO/IEC 13818-2, "Information Technology – Generic coding of moving pictures and associated audio information – Part 2: Video." November 1994

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

[4] ISO/IEC JTC1/SC29/WG11/93-400, "MPEG-2 Test Model 5 (TM5)." Test Model Editing Committee. April 1993

[5] Eric Viscito and Cesar Gonzales. "A Video Compression Algorithm with Adaptive bit Allocation and Quantization," *Proceedings of SPIE The International Society for Optical Engineering: Visual Communications and Image Processing'91*. vol. 1605. part 1. 1991. p.58-71

[6] Atul Puri and R. Aravind. "Motion-Compensated Video Coding with Adaptive Perceptual Quantization," *IEEE Trnasactions on circuits and Systems for Video Technoloby*, vol. 1, no. 4, Dec 1991. p.351-361

[7] Fu-Heui Lin and Russell M. Mersereau. "A Quality Measure-Based Rate Control Strategy for MPEG Video Encoders," *Proceedings of IEEE Internation Symposium on Circuits and Systems*, vol.2. 1996. p.782-783

[8] Wei Ding and Bede Liu. "Rate Control of MPEG Video Coing and Recording by Rate-quantization Modeling," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 1, Feb 1996. p.12-20

[9] King-Wai Chow and Bede Liu. "Complexity Based Rate Control for MPEG Encoder," *IEEE International Conference on Image Processing,* vol. 1. 1994. p.263-267

[10] Gunnar Karlsson. "Asynchronous Transfer of Video," *IEEE Communications Magazine*, v 34 n 8. August, 1996. p.118-126

[11] S. Iai and N. Kitawaki, "Effects of Cell Loss on Picture Quality in ATM Networks," *Elect. and Commun. in Japan*, part 1, vol. 75, no. 10. p.30-41

[12] Pramod Pancha and Magda El Zarki. "Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 3, June 1993. p.190-198

[13] W. Verbiest, L. Pinno, B. Veoten. "Statistical Multiplexing of Variable bit Rate Video Sources in Asynchronous Transfer Mode Networks," *IEEE GLOBECOM'88*, vol. 1. 1988. p.208-213

[14] D. Reininger and D. Raychaudhuri. "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Netowkrs," *IEEE INFOCOM'93*, vol. 3. 1993. p.919-926

[15] R. M. Rodriguez-Dagnino, M. R. K. Khansari, and A. Leon-Garcia. "Prediction of Bit Rate Sequences of Encoded Video Signals," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 3, April 1991. p.305-314

[16] Ajanta Guha and Daniel J. Reininger. "Multichannel Joint Rate Control of VBR MPEG Encoded Video for DBS Applications," *IEEE Trans. on Consumer Electronics*, vol. 4, no. 3. 1994. p.616-623

[17] Gertjan Keesman. "Multi-Program Video Compression Using Joint Bit-Rate Control," *Philips Journal of Research*, vol. 50. no. ½. 1996. p.21-45

[18] Limin Wang and André Vincent. "Joint Rate Control for Multi-Program Video Coding," *IEEE Trans. on Consumer Electronics*, vol. 42, n. 3. August 1996. p.300-305

[19] Sanghoon Lee, Seong Hwan Jang, and Jeong Su Lee. "Dynamic Bandwidth Allocation for Multiple VBR MPEG Video Sources," *IEEE International Conference on Image Processing*, vol.1. 1994. p.268-272

[20] P. N. Tudor and O. H. Werner. "Real-Time Transcoding of MPEG-2 Video Bit Streams," *IEE International Broadcasting Convention*, n 447. 1997. p.298-301

[21] Ekaterina Barzykina. "MPEG Video Coding with Adaptive Motion Compensation and Bit Allocation Based on Perception Criteria," Master thesis. University of British Columbia, Canada. April 1998

[22] Y. Lee. "Rate-Computation Optimized Block Based Video Coding," PhD. thesis. University of British Columbia, Canada. January 1998

[23] ISO/IEC 17211, "Information Technology – Generic coding of moving pictures and associated audio information at 1.5 Mbps."

[24] "Key Dates in Satellite Television." http://www.sbca.com/key_dates.html (July 14, 1998)

[25] "Q&A: DirectTV vs. the Competition." http://www.directv.com/sales/answer_competition.html#prime (July 15, 1998)

[26] Jerry Whitaker. *DTV: The Revolution in Electronic Imaging*. McGraw Hill, New York. 1998

[27] Markus Wasserschaff, Laurent Boch, Rainer Schäfer, Heinz Fehlhammer, Alexander Schertz, and Manfred Kaul. "State of the Art on Video Transmission and Encoding," *Distributed Video Production (DVP) Project A089*. September 1996. http://viswiz.gmd.de/DVP/Public/deliv/deliv.211/delivstr.htm (Jun 29, 1998)

[28] Robert Brehl. "Legal satellite dishes gaining upper hand on grey market," *The Globe and Mail: Report on Business*. July 16, 1998. p.B1&B4

[29] J. A. Flaherty. "Digital TV and HDTV." http://www.web-star.com/hdtvforum/DigitalTVandHDTV.htm (July 16, 1998)

[30] ATSC Doc A/54, "Guide to the Use of the ATSC Digital Television Standard," Advanced Television Systems Committee. Oct 1995. http://www.atsc.org (Feb 20, 1998)

[31] Satoshi Kondo and Hideki Fukuda. "A Real-Time Variable Bit rate MPEG2 Video Coding Method for Digital Storage Media," *IEEE Trans. on consumer Electronics*, vol. 43, no. 3. 1997. p.537-543

[32] Pramod Pancha and Magda El Zarki. "MPEG Coding for Variable Bit Rate Video Transmission," *IEEE Communications Magazine*, vol. 32, n. 5. May 1994. p.54-66

[33] Jiro Katto and Mutsumi Ohta. "Mathematical Analysis of MPEG compression Capability and Its Application to Rate Control," *IEEE Proceedings on Image Processing*, vol. 2. 1995. p.555-558

[34] P. N. Tudor. "MPEG-2 Video Compression Tutorial," *IEE Collloquium on MPEG-2 – What it is and What it isn't*. 1995. p.2/1-2/8

[35] Robert J. Safranek, Charles R. Kalmanek Jr., and Rahul Garg. "Methods for Matching Compressed Video to ATM Networks," *IEEE Proceedings on Image Processing*, vol.1. 1995. p.13-16

[36] Manuela Pereira and Andrew Lippman. "Re-Codable Video," *IEEE Proceedings on Image Processing*, vol. 2. 1994. p.952-956

[37] Wei Ding. "Joint Encoder and Channel Rate Control of VBR Video over ATM Networks," *IEEE Trans. on circuits and Systems for Video Technology*, vol. 7, no. 2. April 1997. p.266-278

[38] "Broadcast video servers are part of MTV Europe's digital future," *HP Telecommunications News*, issue 11. Feb 1997. http://www.tmo.hp.com/tmo/tcnews/9702/11tnfe2.htm (May 14, 1998)

[39] "Testing the MPEG-2 transport stream in DVB networks," *HP Telecommunications News*, issue 11. Feb 1997. http://www.tmo.hp.com/tmo/tcnews/9702/11tnfe5.htm (May 14, 1998)

[40] "NVOD makes the case for delivering enhanced pay-per-view services," *HP Telecommunications News*, issue 11. Feb 1997. http://www.tmo.hp.com/tmo/tcnews/9702/11tnfe3.htm (May 14, 1998)

[41] Craig Birkmaier. "A Visual Compositing Syntax for Ancillary Data Broadcasting." June 1997. http://pcube.com/dtv.html (June 21, 1998)

[42] Cragi Birkmaier. "Limited Vision: The Techno-Political War to Control the Future of Digital Mass Media" http://pcube.com/dtv.html (July 16, 1998)

[43] "The High-Tech Behind Broadcasting DIRECTV," *DIRECTV Hardware – A More Technical Explanation.* http://www.directv.com/hardware/tech.html (May 15, 1998)

[44] Al Kovalick. "Television in Transition: Insights and Solutions for the DTV Broadcaster," *A White Paper on Digital TV.* *http://www.tmo.hp.com/tmo/literature/English/VID_WhitePaper_005a.html* (June 8, 1998)

[45] Saif Zahir, Panos Nasiopoulos, and Victor C. M. Leung, "A High Quality Real Time VBR MPEG-2 System for Broadcasting Applications," *Digest of Technical Papers from IEEE International Conference on Consumer Electronics*, June 1997. p.182-183