ADAPTATION OF VARIABLE-BIT-RATE COMPRESSED VIDEO FOR TRANSPORT OVER A CONSTANT-BIT-RATE COMMUNICATION CHANNEL IN BROADBAND NETWORKS

By

Chi-Yin Tse

A Thesis

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Information Engineering

The Chinese University of Hong Kong

June 1995

Acknowledgement

I first wish to thank Jennifer and my family for their understanding, support and encouragement.

I am most grateful to Dr. Soung Liew for his supervision in these two years.

I also thank Hanford Chan, Kong Huang and Wai-Ho Lee for their technical support.

Abstract

Although widespread digital video transmission is now on the horizon with the introduction of ATM based broadband networks, there are still a number of problems which must be solved before efficient video transmission is possible on broadband networks. Specifically, how to achieve both the advantages of steady image quality and simple transport of video traffic. It turns out that VBR-compressed video sequences should be adapted to CBR by either spatial smoothing or temporal smoothing, so that they can be delivered over the network using a CBR channel.

This thesis studies both spatial smoothing and temporal smoothing for the VBR-CBR video traffic adaptation. For spatial smoothing, we investigate video aggregation, a concept that integrates compression and statistical multiplexing of video information. The shortcomings of previous approaches that separate the processes of compression and multiplexing are explained based on considerations of image quality, bandwidth usage, network management and operation: we argue that it is better to perform compression and multiplexing together before the bundle of video traffic enters the network. We present experimental results which demonstrate the advantages of video aggregation in terms of superior image quality and efficient bandwidth usage.

For temporal smoothing, we establish a framework for video traffic adaptation based on a linear-feedback control model. Important issues of this adaptation scheme, such as stability, robustness against scene change and coding-mode switching, and the trade-off between image-quality and buffer-occupancy fluctuations, are studied with a control-theoretic approach. Compared with previous video traffic adaptation schemes, our framework allows systematic designs and analysis of the adaptation controller. More importantly, this control-theoretic framework may open up many new possibilities for further research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Future broadband communications networks are expected to carry information from a wide variety of services and applications. Video and image traffic, however, is likely to dominate because they are naturally more bandwidth-hungry than other media. It is therefore important to understand how video traffic might best be multiplexed, transported, and switched.

The Asynchronous Transfer Mode (ATM) [1] has been accepted by the International Consultative Committee for Telecommunications and Telegraphy (CCITT) as the basis for multiplexing and switching in future broadband networks. This is mainly due to the high flexibility of ATM networks in handling multimedia services. In ATM networks, data are packetized into fixed-length cells of 53 bytes. Cells are routed in the network independently based on the routing information contained in their 5-byte headers. These cells may be discarded inside the network when traffic congestion occurs.

Unlike general data transmission, video transmission is delay-sensitive: the delay and delay-jitter requirements must be stringent to 1) keep the transmitter

and the receiver synchronized; 2) avoid choppy motion of the received video; and 3) to facilitate interactive control (e.g., in the case of interactive video-on-demand and video-conferencing services). Hence, for video transmission, re-transmission of the discarded cells is usually not allowed. Although video to some extent is noise-tolerant, cell loss may still result in serious image-quality degradation (to be discussed later).

Therefore, although widespread digital video transmission is now on the horizon with the introduction of ATM based broadband networks, there are still a number of problems which must be solved before efficient video transmission is possible on broadband networks.

This chapter gives a brief introduction to our research focus. Section 1.1 describes video compression and transport in a generic manner. Section 1.2 gives motivations for the VBR-CBR adaptation of video traffic. The research problems we are interested in are described in Section 1.3. Lastly, Section 1.4 describes the organization of the remainder of this thesis.

1.1 Video Compression and Transport

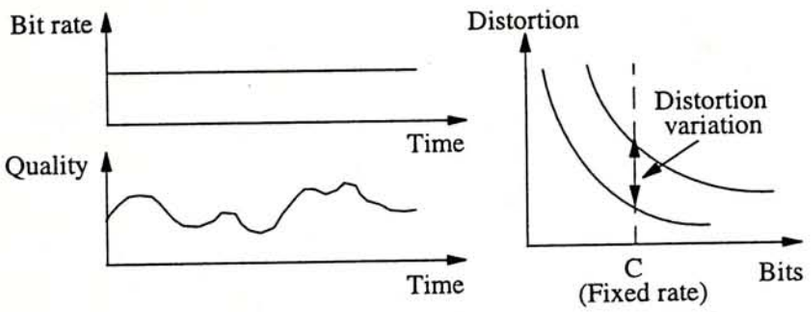
To reduce the bandwidth needed, video is almost always compressed before transmission. Video compression schemes can be classified into variable bit-rate (VBR) or constant bit-rate (CBR) compression, according to whether the output rate of the encoder is variable or constant [2].

For a given video stream, its intrinsic bandwidth requirements may vary over time as to maintain a constant image quality, thanks to the scene content variations of the underlying video sequence. In CBR compression, the output

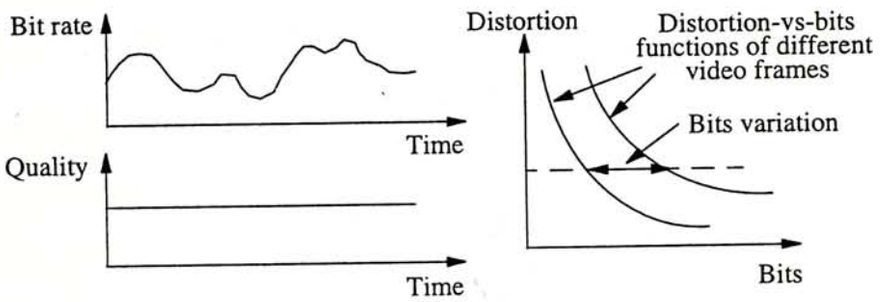
bit rate of the encoder is forced to be constant. Because scenes that intrinsically demand high bandwidths may have their bandwidths cut down to maintain the constant output bit rate, the image quality varies over time (Fig. 1.1 (a) [2]). In contrast, in VBR compression, the output bit rate of the encoder varies over time according to the intrinsic bandwidth requirements of the underlying video sequence. Therefore, the image quality is more or less constant (Fig. 1.1 (b)). Moreover, in order to enhance the compression performance, and to support some important functions (e.g., fast-forward display), some VBR compression schemes use different coding modes, which provide different compression ratio and features, to code the frames. For instance, the Moving Picture Experts Group (MPEG) compression scheme [3, 4, 5] switches over intraframe, interframe and interpolative coding modes periodically. Since VBR compression schemes can provide higher averaged compression ratio and more steady image quality, they are generally preferred to CBR compression.

As a matter of fact, compression schemes that lie somewhere between the two extremes (i.e., fixing the output bit rate and fixing the image quality) are also possible. In general, in the consideration of compression, there is a kind of "uncertainty principle" which consists of the tradeoff between the variations of bit rate and image quality.

CBR and VBR transport, as distinct from compression, refers to using constant-bit-rate and variable bit-rate channels, respectively, for the transport of data. Transport using CBR channels has many advantages from the networking viewpoint. Since the data rate is constant, bandwidth allocation and tariff for network usage are simple. It is also straightforward for the network to multiplex several CBR channels onto a common physical link and guarantee the delivery



(a) CBR video compression.



(b) VBR video compression.

Figure 1.1: Characteristics of VBR and CBR video compression schemes.

of all information since traffic arrives at a predictable rate.

It is natural to use CBR transport for CBR-compressed data. Similarly, VBR compression followed by VBR transport is a natural combination. In the second case, however, it is difficult to multiplex VBR video streams while guaranteeing the delivery of all cells, because the multiplexed streams may all output a large number of bits simultaneously. Since video data have been highly compressed, when cell loss occurs, and especially when some important data (e.g., header information and grey-level signals of the images) are contained in the discarded cells, serious image-quality degradation may result. Generally, more than the average bandwidth needs to be allocated to a VBR stream to maintain a small cell-loss probability. Even then, absolute cell-delivery guarantee is not possible unless the peak bandwidth is allocated, in which case the delivery of the VBR stream will be expensive. For public networks, the fact that cells may be dropped due to interference from other streams also complicates the tariff problem and the contractual agreement between the network operator and user.

Two-layer coding and transport strategy [2, 5, 6, 7, 8] is proposed to alleviate the image-quality degradation problem of transport of VBR compressed video. However, it complicates the network operation and decoder design (this shall be discussed in section 2.3), as well as may result in blocky effect on reconstructed images [4].

1.2 VBR-CBR Adaptation of Video Traffic

Given the desirability of VBR compression (which offers relatively constant image quality) and CBR transport (which facilitates simple network operation), an

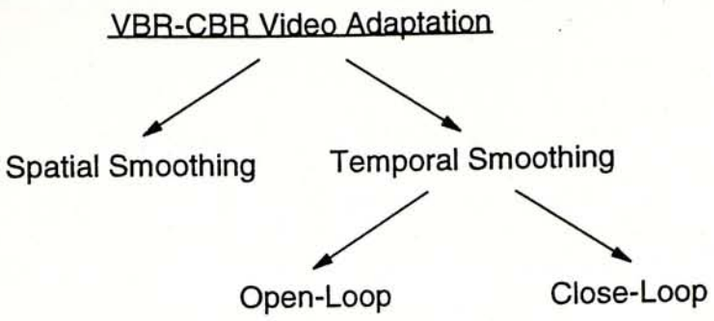


Figure 1.2: A possible taxonomy for different VBR-CBR video adaptation approaches.

issue is how to adapt VBR-compressed video traffic to CBR channels. In other words, how to smooth the traffic generated by VBR video encoders so that it can be delivered over the network using a CBR channel. It turns out that this VBR-CBR adaptation of video sequences can be achieved with several possibilities. Figure 1.2 is a possible taxonomy for different VBR-CBR adaptation approaches.

In spatial smoothing, bandwidth of a CBR channel is statistically shared among a number of VBR-coded video streams: betting that not all the video streams will demand high bit rates simultaneously, the bandwidth-hungry streams may "borrow" bandwidth from the less demanding streams. Although individually the streams are VBR, as a whole the traffic is adapted to be CBR.

In temporal smoothing, bandwidth of a CBR channel is shared among different time moments of a VBR-coded video stream. Data from the output of the VBR encoder are fed to a smoothing buffer, which then forwards data at a constant rate to the network. This method is especially effective in smoothing the different bandwidth requirements of successive frames.

VBR-CBR adaptation by temporal smoothing can be further divided into

two classes, according to whether the VBR-compression and the temporal smoothing are in an open-loop or in a close-loop. For the former case, the VBR-encoder compresses the video sequence at a constant image quality, regardless of the buffer occupancy at the smoothing buffer. In contrast, for the close-loop case, the VBR-encoder adapt its output bit rate (by coding the video sequence at different image quality) in response to the occupancy of the smoothing buffer.

In order to prevent cell loss due to buffer overflow at the smoothing buffer (which may results in serious image quality degradation of the reconstructed video), for temporal smoothing, the close-loop case usually is preferred to the open-loop case. Therefore, when we are talking about temporal smoothing in this thesis, we simply refer the close-loop case by "adaptation" unless specified.

Note that spatial and temporal smoothing are complementary, in the sense that in a hybrid system, a number of video streams can first be smoothed to less bursty (by spatial smoothing), afterwards, this multiplexed traffic is further smoothed by temporal smoothing.

1.3 Research Contributions

This thesis studies both spatial and temporal smoothing for the adaptation of VBR-coded video traffic to CBR transport over a CBR channel in broadband networks. The research consists of two parts: the first part concerns spatial smoothing among different video streams, and the second concerns temporal smoothing within a single video stream.

1.3.1 Spatial Smoothing: Video Aggregation

Spatial smoothing of video traffic has been studied by many researchers [9, 10, 11, 12, 13], however, most of them assumed that the smoothing process (i.e., statistical multiplexing) and the compression processes at the video encoders are independent. One consequence is that serious image quality degradation may result due to cell loss at the multiplexing buffer.

Knowing that we can improve the image quality of the video sequences by dropping data selectively according to their relative importance when needed, we suggest to perform compression and multiplexing together before the traffic enters the network: when the assigned bandwidth is exceeded, data are dropped selectively according to their relative importance. We call this integration of compression and statistical multiplexing as *aggregation*. We shall demonstrate the advantages of video aggregation in terms of image quality, bandwidth usage, network management and operation.

1.3.2 Temporal Smoothing: A Control-Theoretic Study

Unlike in spatial smoothing, many previous researchers [14, 15, 16, 17] indeed suggested that video compression and temporal smoothing of video traffic should work collaboratively in a close loop. However, most of these previous adaptation schemes are somewhat *ad hoc* in nature, and their operation is not analyzable.

In contrast, we study the video traffic adaptation system in a control-theoretic framework. We model the adaptation system as a linear-feedback system, and analyze its stability and other performance issues in a quantifiable manner. Compared with previous schemes, our approach allows systematic designs and

analysis of the adaptation controller. More importantly, the framework opens up many new possibilities for further research.

While most of our discussions on video aggregation and traffic adaptation are applicable to generic VBR video compression schemes, implementations of the proposed concepts will be based on the MPEG compression standard.

1.4 Organization of Thesis

The remainder of this thesis is organized as follows. Chapter 2 reviews MPEG compression, problems of transmitting an MPEG video stream, and two-layer coding and transport strategy as preliminaries to our latter study. Chapter 3 proposes a novel scheme for spatial smoothing of video traffic, called *aggregation.* Aggregation can be treated as the integration of data compression and statistical multiplexing. Chapter 4 study the generic temporal smoothing and encoder bit rate adaptation problem with a control-theoretic approach. Finally, Chapter 5 gives conclusion to the thesis. Each of these chapters contains a bibliography for guilding readers who want to study a particular topic more in detail.

Bibliography

[1] M. De Prycker, *Asynchronous Transfer Mode : Solution for Broadband ISDN*, Ellis Horwood, 1993.

[2] N. Ohta, "Packet Video : Modeling and Signal Processing", Artech House. 1994. pp. 164.

[3] D. Le Gall, "MPEG : A Video Compression Standard for Multimedia Applications", *Commun. of the ACM*, Vol. 34, pp.47–58, April 1991.

[4] C-Y Tse and S. C. Liew, "Video Aggregation : An Integrated Video Compression and Multiplexing Scheme for Broadband Networks," *Proc. IEEE Infocom '95.*

[5] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Commun. Magazine*, May 1994.

[6] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 3, June 1993.

[7] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks", *IEEE J. Selected Areas in Commun.*, Vol. 7, No. 5, June 1989.

[8] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM Networks", *IEEE J. Selected Areas in commun.*, Vol. 7, No. 5, June 1989.

[9] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta and J. Hill, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks", *Proc. IEEE Infocom 93*, pp. 919–926.

[10] S. S. Dixit and P. Skelly, "Video Traffic Smoothing and ATM Multiplexer Performance", *Proc. IEEE Globecom 91*, pp. 0239–0243.

[11] R. Coellco and S. Tohme, "Video Coding Mechanism to Predict Video Traffic in ATM Network", *IEEE Globecom 93*, pp. 447–450.

[12] D. M. Cohen and D. P. Heyman, "Performance Modeling of Video Teleconferencing in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 6, December 1993.

[13] N. M. Marafih, Y-Q. Zhang and R. L. Pickholtz, "Modeling and Queuing Analysis of Variable-Bit-Rate Coded Video Sources in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 4, No. 2, April 1994.

[14] L. W. Lee, J. F. Wang, J. Y. Lee and C. C. Chen, "On the Error Distribution and Scene Change for the Bit Rate Control of MPEG", *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 3, August 1993.

[15] C-T Chen and A. Wong, "A Self-Governing Rate Buffer Control Strategy for Pseudoconstant Bit Rate Video Coding," *IEEE Transactions on Image Processing*, Vol 2, No. 1, Jan 1993, pp. 50–59.

[16] L. Wang, "Bit Rate Control for Hybrid DPCM/DCT Video Codec", *IEEE Trans. Circuit and Systems for Video Tech.*, Vol. 4, No. 5, October 1994.

[17] H. Watanabe and S. Singhal, "Bit Allocation and Rate Control Based on Human Visual Sensitivity for Interframe Coders", *Proc. ICASSP 1992.*

Chapter 2

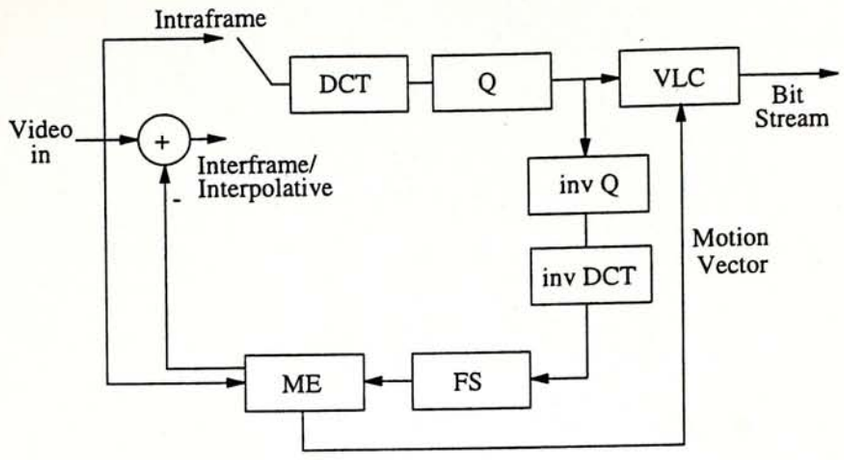
Preliminaries

Before going into the details of our research on smoothing of VBR-coded video traffic, let us first review the basics of MPEG compression, the problems of transmitting an MPEG-coded video sequence, as well as previously proposed two-layer coding and transport strategy.

2.1 MPEG Compression Scheme

The MPEG compression [1, 2, 3] has been developed as a standard of video compression and is fast becoming the coding scheme of choice among product developers.

The schematic of an MPEG coder is shown in Fig. 2.1 [1]. In the MPEG coding standard, spatial information of a frame is partitioned into four layers: frame, slice, macroblock and block. A frame is the basic unit of display, and is further divided into slices. A slice is a sequence of macroblocks. A 16×16 (16 pixels by 16 pixels) macroblock (MB) is the unit for motion compensation, and



DCT	: Discrete Cosine Transform	inv DCT	: inverse DCT
Q	: Quantization	inv Q	: inverse Quantization
FS	: Frame Storage	ME	: Motion Estimation
VLC	: Variable Length Coding		

Figure 2.1: Schematic MPEG coder.

it consists of 8×8 blocks. Discrete cosine transform (DCT) is performed on each 8×8 blocks. For color video, an MB consists of four 8×8 luminance blocks and two 8×8 chrominance blocks. For each frame, there are three choices of coding algorithms: Intraframe, interframe and interpolative coding.

Intraframe-coded frames (I) are coded independently, and therefore can act as refreshments of accumulated error (as we will see), as well as access points for random access of the video sequence. The whole I frame undergoes 8×8 block-based DCT without referring to other frames. The DCT coefficients are then quantized. The quantization matrix ensures that the lower frequency components are quantized with smaller step size, while higher frequency components have coarser quantization (we will discuss this later). Before variable-length

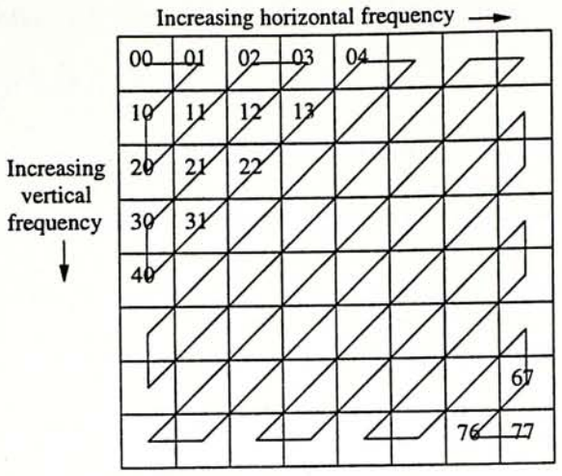


Figure 2.2: Zigzag scanning order of DCT components.

coding (VLC), the DC coefficients of individual blocks are first coded differentially within the corresponding slices. This is because DC coefficients (which correspond to the grey level) of adjacent blocks are usually strongly correlated. For VLC, each non-zero AC component is first grouped with the runlength of preceding zero components (in zig-zag order, see Fig. 2.2), and then assigned a codeword from a Huffman table.

For interframe-coded frames (P), temporal redundancy is first reduced by causal MB-based motion compensation, with respect to the preceding I or P frames stored in the Frame Storage. If the motion estimation (ME) error for a MB is less than a threshold (i.e., there is enough redundancy such that interframe coding is worthwhile), then the motion vector (MV) will be differentially and then VLC coded, while the ME error will undergo DCT, coarse quantization and then VLC. Otherwise, that MB will undergo intraframe coding. Interpolative frames (B) are coded in a way similar to coding P frames; however, the motion compensation is bi-directional with respect to both the preceding and following P (or I) frames.

Although interframe and interpolative coding generally can provide higher compression ratio, intraframe coding is still used by the coder periodically. It is because when some data in I and P frames are lost during transmission, the frame contents in the Frame Storages at the coder and decoder become different. Even if no further data is lost, for the following P and B frames, the ME at the coder and decoder will refer to different frame contents as the "baseline" of estimation. Consequently, errors due to data loss of one I or P frame will propagate along the following P and B frames, and this is often referred to as "error propagation". However, the accumulated errors can be cleared by sending an I frame. Moreover, by introducing I frames periodically in the video sequence, some preferred display functions (e.g., fast-forward, reverse display, etc.) can be facilitated.

A Group of Pictures (GOP) is defined as the smallest group of frames which shows the coding-mode switching pattern of an MPEG encoder. A GOP can be characterized by two parameters, N and M: $N-1$ is the number of frames coded between successive I frames, while $M-1$ is number of B frames coded between successive P frames. A GOP (in display order) with $N = 10, M = 3$ is as follows.

IBBPBBPBBP

Note that the encoding order of frames is different from the display order. It is because in order to facilitate bi-directional motion compensation, a B frame have to be coded after the P frame which follows it in the display order. In other words, when interpolative coding is employed, the MPEG encoder will introduce an extra delay of M frame-periods due to re-ordering the input frames [4].

For an MPEG encoder with given values of N and M, the output image-quality and bit rate can be adjusted by the quantization factor, Q. Quantization factor Q controls the degree of fineness of quantization by scaling the elements in the quantization matrix: when Q is increased, the quantization of the DCT coefficients becomes coarser.

Note that data in an MPEG-coded video stream are of unequal importance. The header information, MV's and DC components are obviously very important. Among the DCT AC components, those of lower frequencies are more important than those of higher frequencies for two reasons. First, the energy (i.e., amplitude square) of the DCT AC components tends to decrease along the zig-zag scanning order (i.e., energy compaction) [5]. Second, human vision system is less sensitive to the high frequency signals.

2.2 Problems of Transmitting MPEG Video

As mentioned in Chapter 1, the output bit rate of a VBR-encoder varies over time, because of 1) variations of scene contents of the underlying video sequence, and 2) switching of coding-modes. In order to see how the VBR-coded video traffic may vary, the traffic in terms of bits per frame of one MPEG-coded video sequences is shown in Fig. 2.3. (The resolution and frame rate of the sequence are 320×240 and 30 frames per second, respectively.) Note that sharp peaks occur periodically because of intraframe coding. In addition, the local average rate of the traffic keeps varying over time, thanks to the scene content variations of the frames. In general, the traffic of MPEG coded sequences is rather bursty.

It is difficult to transport and multiplex such bursty traffic in broadband

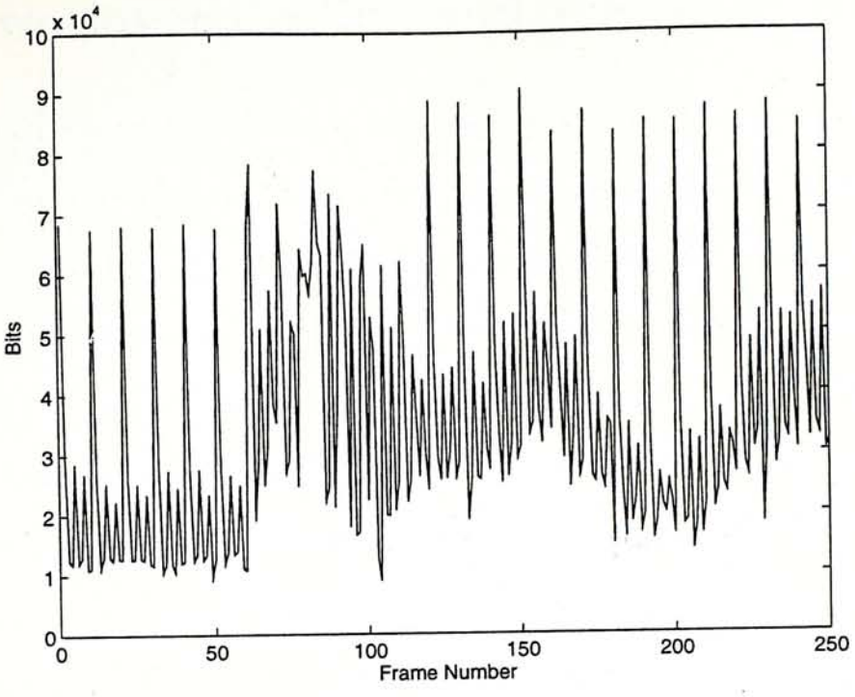


Figure 2.3: Bits per frame of an MPEG-coded sequence

networks while guaranteeing the delivery of all cells, unless bandwidth corresponding to the peak rate of the traffic is reserved. When one cell is lost, all data of the consecutive MB's contained in it (in general, one to ten MB's) are lost [6]. Moreover, in a congested network, cells loss tends to occur in bursts. Hence, occurrences of cell loss will usually result in large areas of poor-quality clusters in the reconstructed images. Furthermore, this serious degradation will propagate along P and B frames and hence last for some time. In fact, if some of the header information of the MPEG-coded video stream is lost, the decoder may be even unable to decode the received data.

2.3 Two-layer Coding and Transport Strategy

In order to alleviate the image-quality degradation problem of transmitting VBR-coded video traffic, two-layer video coding and transport strategy has been widely investigated [1, 6, 7, 8, 9, 10, 11]. In the two-layer approach, VBR-coded data of a video stream are divided into the base-layer that contains the basic-quality-image data and the second-layer that contains the image-enhancement data. The base-layer is transported as the guaranteed stream (GS) and the second-layer is transported as the enhancement stream (ES). The idea is that only the data in the enhancement stream can be dropped.

2.3.1 Framework of MPEG-based Layering

There are mainly two approaches for MPEG-based layering [1, 6, 7, 8, 9]: bit-plane and frequency-plane [8]. In bit-plane layering, the DCT coefficients are coarsely quantized into base-layer first, while the second-layer codes those quantization errors [6]. The boundary separating the two layers is controlled by the quantization factor for coding the base-layer. Thus, the larger quantization factor is used for the base-layer, the less information will be assigned to the base-layer. In frequency-plane layering, lower frequency components are assigned to the base-layer, while the higher ones are assigned to the second-layer [1, 9]. Therefore, the boundary between the two layers is the number of lower frequency components assigned to the base-layer. In both approaches, all the headers, MV's and DC components are assigned to the base-layer.

2.3.2 Transmission of GS and ES

Two-layer video coding and transport strategy requires rate control of the GS: otherwise, the network can hardly guarantee delivery of cells once the pre-negotiated rate is exceeded. To do so, the boundary separating the base-layer and the second-layer can be fixed so that the bit rate of the GS will never exceed the reserved bandwidth (e.g., token generation rate of the leaky bucket [12]). Alternatively, the boundary can be adjusted dynamically in response to the buffer-occupancy at the user-network-interface (UNI) [12]. While the former approach is simpler to implement, the latter approach can potentially achieve a better basic image quality.

In contrast, transmission of the ES is considered to be some kind of non-guaranteed service, which needs no rate control nor policing at the UNI. The ES cells may arrive at the destination host, or be discarded inside the network when they come across traffic congestion.

2.3.3 Problems of Two-layer Video Transmission

Although two-layer video coding and transmission can alleviate the image-quality degradation problem due to cell loss, the transmission aspect presents some problems. If both the GS and the ES are transmitted by a single virtual channel (VC) [12], the network control becomes complicated because it has to handle cells of two priorities in a single VC. Although there is one priority bit in the ATM-cell header that can be used for this purpose [12], it is not clear that it is best use to distinguish video data this way. On the other hand, if the two streams are transmitted by two different VC's, the video session may have synchronization

problem, as the data transmitted by the two streams may experience different delay and delay jitter.

A second problem is operational in nature. Suppose that two-layer transmission is used on a public network, then what is the incentive for the public-network operator to deliver the ES cells? If there need not be any guarantee, then it behooves the network operator 1) to focus on better performance guarantee for the GS stream and 2) to use any extra network bandwidth for other services. Most schemes studied to date assume that ES cells are discarded only when a multiplexer's buffer begins to overflow. In other words, when buffer does not overflow, the ES streams may still interfere with the GS streams or cells of other services and influence their delay performance. Thus, the network will be able to perform its *pre-negotiated job* better by simply dropping the ES cells at the UNI without even attempting to transmit them. Otherwise, a greedy user that replicates ES cells to increase the chance of delivery will grab all the excess network bandwidth, to the detriment of other users. To prevent this, one may argue that the user must also pay for the delivery of the ES cells and, at the same time, demands a weaker form of guarantee from the network. But what is the degree of guarantee, how bandwidth should then be allocated and ES streams be policed become an even tougher issue to address.

Moreover, from the decoding viewpoint, more processing is needed at the receiver to combine data from the two layers before decoding.

Bibliography

[1] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Commun. Magazine*, May 1994.

[2] D. Le Gall, "MPEG : A Video Compression Standard for Multimedia Applications", *Commun. of the ACM*, Vol. 34, pp.47–58, April 1991.

[3] G. K. Wallace, "The JPEG Still Picture Compression Standard", *IEEE Trans. on Consumer Electronics*, Vol. 38, No. 1, February 1992.

[4] M. Kawashima, C. Chen, F. Jeng and S. Singhal, "Adaptation of the MPEG Video-Coding Algorithm to Network Applications", *IEEE Trans. Circuit and Systems for Video Tech.*, Vol. 3, No. 4, August 1993.

[5] K. R. Rao and P. Yip, *Discrete Cosine Transform : Algorithm, Advantages, and Applications*, Academic Press, Inc., 1990, pp. 170.

[6] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 3, June 1993.

[7] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks", *IEEE J. Selected Areas in Commun.*, Vol. 7, No. 5, June 1989.

[8] N. Ohta, "Packet Video : Modeling and Signal Processing", Artech House. 1994. pp. 164.

[9] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM Networks", *IEEE J. Selected Areas in commun.*, Vol. 7, No. 5, June 1989.

[10] W. Verbiest, L. Pinnoo and B. Voeten, "The Impact of the ATM Concept on Video Coding", *IEEE JSAC*, Vol. 6, No. 9, December 1988.

[11] H. Gharavi and M. H. Partovi, "Video Coding and Distribution over ATM for Multipoint Teleconferencing", *Proc. IEEE Globecom 1993.*

[12] M. De Prycker, *Asynchronous Transfer Mode : Solution for Broadband ISDN*, Ellis Horwood, 1993.

Chapter 3

Video Aggregation

This chapter investigates video aggregation, a concept that integrates compression and statistical multiplexing of video information for transport over a communication network. We focus on the scenario where information from a group of video sessions are to be delivered as a bundle to one or more destinations. The shortcomings of previous approaches that separate the processes of compression and multiplexing are explained based on considerations of image-quality, bandwidth usage, network management and operation: we argue that it is better to perform compression and multiplexing together before the bundle of video traffic enters the network. We demonstrate the advantages of video aggregation in terms of image-quality, bandwidth usage, network management and operation.

Application areas of aggregation include video broadcast and video-on-demand. Video programs are transported as a bundle from the video server directly to the subscribers in the former (Fig. 3.1 (a)), and to a distribution node close to the subscribers in the latter [1] (Fig. 3.1 (b)). Aggregation may also find use in the

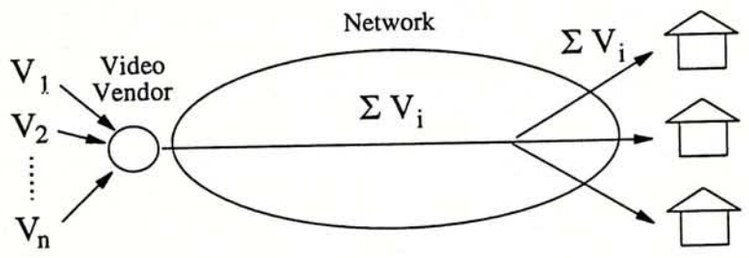
transport of long-distance video-phone data: video streams from various subscribers targeted for a common remote area may be aggregated at a local central office before being delivered as a bundle to the remote central office serving the area (Fig. 3.1 (c)).

3.1 Motivation and Basic Concept of Video Aggregation

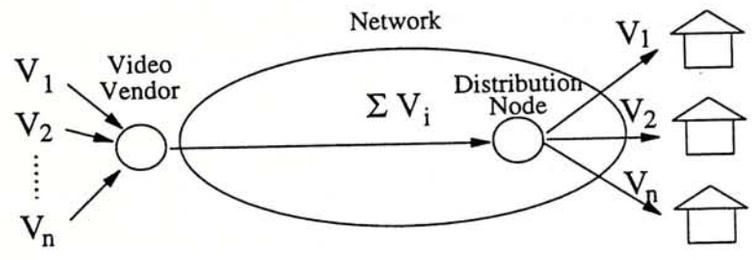
To combine VBR-coded video streams onto the common CBR channel, the straightforward method is to first packetize the output data of the VBR encoders into cells and then multiplex the cells statistically [2, 3, 4, 5, 6]. A problem, however, is that cells may still be dropped due to buffer overflow at the multiplexer, particularly when the VBR streams all output high bit rates simultaneously. When cell loss occurs, and especially when some important data (e.g., header information and grey-level signals of the images) are contained in the discarded cells, serious image-quality degradation may result.

One alternative is to apply the two-layer approach (see Section 2.3) to the video-bundle scenario as follows. The GS's of all video sources are transmitted and they use up certain amount of the bandwidth of the reserved CBR channel. The remaining bandwidth is then used in the statistical multiplexing of the ES's where loss may occur [7]. The multiplexing can be performed by the access network node (i.e., the first node to which the ES's are sent).

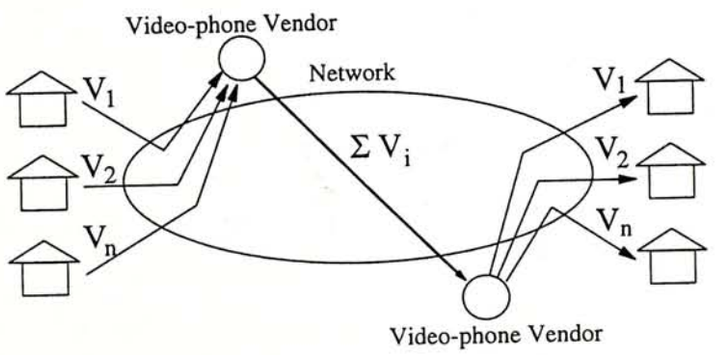
The two-layer approach alleviates the image-quality degradation problem somewhat, but it complicates network operation and decoder design (see Section 2.3).



(a) Video broadcasting.



(b) Video-on-demand.



(c) Long-distant video-phone

Figure 3.1: Applications of video aggregation.

Another shortcoming of the two-layer approach is that it does not achieve the optimal image-quality: the relative importance of the data within an ES and among the separate ES's is not distinguished in the multiplexing process. Typically, within a video stream, not all data are equally important. Also, among the video streams, some streams may require more bandwidth (therefore their ES's contain more signals) than the other streams at a particular moment in time.

In the video-bundle scenario, we have the alternative of multiplexing data before they have been packetized. The advantage of this is that the relative importance of the data is known to the last detail, and one can choose to drop the least significant data when the reserved bandwidth is exceeded. We can potentially achieve 1) better and smoother image-quality for the frames within a video stream , and 2) fairness of image-quality among the video streams. This is the basic observation that motivates video aggregation: integration of video compression and multiplexing into a single process prior to data packetization.

In video aggregation, video sequences are compressed collaboratively such that 1) the sum of the coded bit rates of the video sequences is almost equal to (but not larger than) the reserved bit rate of the CBR channel, 2) within each video stream, data dropped are less important than those retained, and 3) different video streams have roughly the same image-quality according to some signal-to-noise or distortion metric. In the following section, video aggregation is described in an abstract manner as a lossy compression process that is applied after a preliminary compression process.

3.1.1 Description of Video Aggregation

In many video compression schemes, the output data can be divided into segments. Each segment has a certain number of bits, some of which can be dropped, if needed, at the expense of image-quality. Associated with each segment is a function relating the number of bits retained and the corresponding image-quality. Within each segment, bits can be ordered according to their significance so that those of lower significance will be dropped first when necessary.

As an illustration, in MPEG coding (see Section 2.1) the segments could be "blocks" and the bits are from codewords representing the nonzero frequency components in the blocks. The bits in a block can be ordered according to frequency because the codewords of low frequencies are generally more significant to image-quality (see Section 2.1).

In aggregation, a number of segments from each video source is collected. Let n be the number of video streams and k be the number of segments taken from each stream for aggregation. Then, $m = nk$ is the total number of segments collected from all sources. Let B_t be the number of bits reserved for sharing among the m segments (i.e., B_t should be proportional to the reserved bandwidth on the CBR channel). When B_t is insufficient to accommodate all bits of the segments, for each segment i, we compute $B_i(D)$, the number of bits that must be retained in order to maintain a distortion level of D. Note that $B_i(D)$ is computed as a function of D. To select a specific but common operating distortion level for all segments, we find a distortion level D' such that

$$B_1(D') + B_2(D') + \cdots + B_m(D') = B_t \tag{3.1}$$

For each of segment i, the least-significant bits are then dropped so that the

number of bits remaining is $B_i(D')$. In practice, it may not be possible to achieve absolute equality of distortion levels because of the discrete nature of the bits or groups of bits (e.g., codewords) that are dropped. In this case, the aim is to transport no more than B_t bits and to minimize the difference between distortion levels of any two segments.

3.2 MPEG Video Aggregation System

The preceding sections described video aggregation as a generic concept, and is independent of the underlying video compression scheme. This section, in contrast, focuses on the implementation and system design of video aggregation for MPEG-coded video sequences.

3.2.1 Shortcomings of the MPEG Video Bundle Scenario with Two-Layer Coding and Cell-Level Multiplexing

In Section 3.1, we have claimed that the shortcoming of the cell-level multiplexing with two-layer coding and transport is that there is no distinguishing between the relative importance of data within an ES and among the separate ES's. Let us now examine its implications for image-quality in more detail.

1. Blocky Effects within a Frame

In multiplexing ES's, the discarding of an ES cell means that those MB's corresponding to this cell (in general, one to ten MB's [8]) will have only their

base-layer data transmitted. Therefore, only basic image-quality can be provided. In contrast, those MB's having their second-layer data transmitted will provide perfect image-quality. Therefore, unless all cells from a frame can be transmitted, the MB's within a frame will have different qualities due to the discarding of some ES cells and the retaining of others. This results in *blocky effects* on the reconstructed image (image appears as clusters).

2. Non-optimal Image Quality within a Frame

Even though the ES data of a video sequence are of different importance, when they are packetized in cells, there is no further prioritization among them. However, an ES cell is either dropped or transmitted in its entirety. We cannot, say, drop part of an ES cell and part of another ES cell so as to ensure that the missing data are the least significant. As a result, optimality cannot be achieved because some of the dropped data may potentially contribute more to the quality of the reconstructed images than those retained.

3. Fairness of Image Quality among the Video Sequences

Consider the video streams that are multiplexed. To provide the same image-quality, different scene contents may demand different bit rates: video stream A may need more bandwidth than video stream B sometimes, and the reverse may be true at other times. When cells must be dropped at the multiplexer, the multiplexer does not have the knowledge of the significance levels of the ES cells. It is possible that some images (or portions of an image) suffer more visual degradation than others, even if they incur the same cell-loss rate.

The problem of the lack of a measure of the signal degradation due to cell

loss is further compounded by the fact that the importance of the cells varies from intraframe to interframe coding. For example, a cell from an I frame may carry 5% of the signal of the reconstructed image; however, one from a P frame may carry 5% of the ME error, which contributes to only 0.5% of the overall image signal. Certainly, dropping a cell from a P frame is more tolerable than dropping one from an I frame. The multiplexer for two-layer transport does not generally distinguish between P-frame and I-frame cells.

3.2.2 MPEG Video Aggregation

There are several approaches to the implementation of the video-aggregation concept on MPEG-coded video sequences. This section briefly describes the one we used for our experiment, and some alternative approaches are discussed in the appendix (Section 3.6).

The goal of MPEG video aggregation is to ensure that all MB's contained in the corresponding spatial unit (slice or frame) from all video sequences provide more or less the same image quality. In our implementation, the video-aggregation process is slotted into slice periods.[1] In every slice period, data for a slice (which are still in the form of VLC codewords and not yet packetized) is collected from every video sequence. A number of bits are allocated for all the slices to be aggregated. All the header information, MV's, as well as the first β codewords from every 8×8 block are forwarded. This uses up a certain amount of bandwidth. The remaining codewords are then subjected to aggregation with the remaining bandwidth B (note that B may change from aggregation period

[1] In general, the unit of aggregation can be smaller or larger than a slice, depending on the processing capability.

to aggregation period).

There are two reasons why we might want to exempt the first β codewords from the aggregation process. The first reason is that this will reduce the amount of data to be aggregated and hence the complexity of the process. The second reason, which is more subtle, is that this exemption might be advantageous in some variations of aggregation systems (see Section 3.3). Note that small β implies a higher degree of bandwidth sharing among video sessions, and higher bandwidth efficiency can be achieved. There is, however, less guarantee on each individual session. When β is large, there is less sharing, and to the extent that β is large enough, there could be no bandwidth sharing at all. In this case, there will be no significant difference between aggregation and multiplexing.

At the beginning of the aggregation process, the distortions of all MB's with only the DC and first β AC components sent are calculated. The MB that has the lowest image quality is identified. If there are remaining bits, the next codeword from all the 8×8 blocks contained in this MB will be forwarded. The distortion of that MB will then be updated. Afterwards, the next MB that has the lowest image quality is identified and the step is repeated until all the allocated bits for that slice period have been exhausted.

Note that because the codewords for each 8×8 block are arranged with their DCT components in the zig-zag order (see Fig. 2.2 in Section 2.1), for each block, the codewords discarded during aggregation are of higher frequencies and hence are less important.

The signal-to-noise ratio (SNR) is commonly used as an objective measurement for image quality. However, the actual signal energy of an MB from P or B frames can be found only after it has been decoded (with respect to the

reference frame) back into the spatial domain. This is because the reference area (16×16 in size) for motion compensation of an MB is not necessarily fall on the boundary of the MB's. Therefore, unless the signal energy in each MB is provided by the MPEG encoders, using SNR as the metric for image quality during aggregation is not feasible (unless, of course, the aggregator decodes the MPEG sequences to find out the signal energies of MB's). Alternatively, we may use noise energy as the metric. Since the amount of energy carried by a codeword is equal to the amplitude square of its non-zero DCT component, the noise energy in an MB during the aggregation process is equal to the sum of the energies of the discarded (or not-yet-sent) codewords.

3.2.3 MPEG Video Aggregation System Architecture

We now look at the overall architecture of the MPEG video aggregation system (VAS). An MPEG VAS comprises a group of MPEG video sources, a VAS server and the ATM Adaptation Layer (AAL) [9] (see Fig. 3.2).

Video sequences are coded independently by MPEG coders with high quality. The coded data are then forwarded to the VAS server (without packetization). The VAS server is responsible for aggregating the video sequences, as well as reassembling the forwarded data block by block after aggregation. If the codewords in the video sequences have been Huffman-coded, the VAS server should also Huffman-decode them first before performing aggregation.

At the AAL, data of the same video sequence are packetized into cells, and cells from all sequences are transported by either a virtual channel (VC) or a dedicated CBR virtual path with different VC's for different sequences.

In principle, the allocated number of bits for a slice period can either be fixed

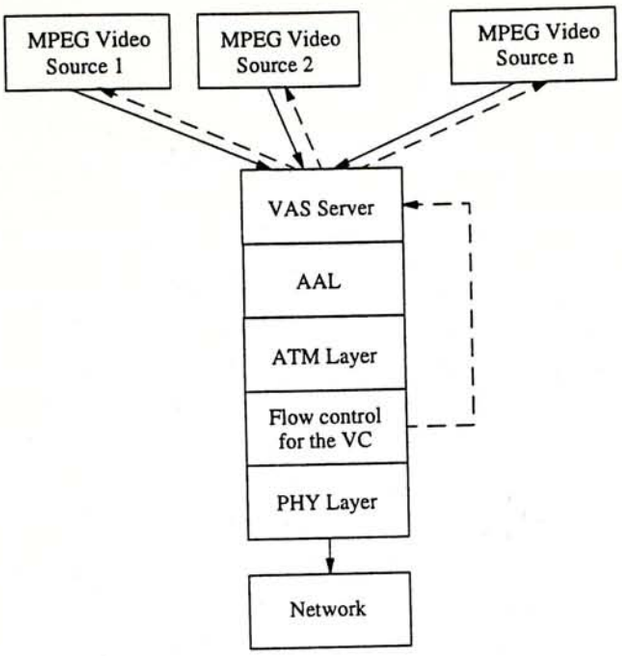


Figure 3.2: Schematic diagram of an MPEG VAS. Solid arrows show the flow of data, while dotted arrows show feedback (if any).

or varied. In the first case, the output from the VAS enters the CBR channel of the network directly. Temporal statistical multiplexing (i.e., smoothing of traffic generated at different time instants) is confined to a slice period only. In the second case, the output enters a buffer which in turn outputs data at a constant rate to the network; the allocated number of bits to a slice (rate control) varies according to the state of the buffer-occupancy. The second case allows for smoothing of traffic over a longer time period as compared to the first case, at the expense of more complicated operation and additional delay jitters at the buffer.

Note that with aggregation, the network will no longer need to bother with VBR video traffic or priorization of cells. In addition, aggregation has the advantage of being transparent to both MPEG encoders and decoders. At the

transmitter side, to the extent that aggregation is introduced as an add-on process, standard MPEG encoders can be used. Stored video sequences, which are previously compressed by standard MPEG encoders, do not have to be decompressed and then re-compressed during the aggregation process. The forwarded data can be easily put into the standard MPEG format after aggregation. Therefore, at the receivers, standard MPEG decoders can be used without the need for any add-on equipment (*cf.* with two-layer approach, the receivers must combine the two layers before decoding). This is an especially attractive feature considering that in many video-distribution systems, there could be many receivers to each transmitted video stream.

3.3 Variations of MPEG Video Aggregation System

During aggregation, some codewords of I or P frames may be discarded because of bandwidth shortage. This may cause error propagation. According to how error propagation is dealt with, MPEG VAS's can be categorized into three classes.

For the partial-reference VAS, only the data of the first β codewords, which are not subjected to aggregation, are put back into the Frame Storages of the coder and decoder as the reference for interframe and interpolative coding/decoding. Since the delivery of these data is guaranteed, error propagation will not occur. However, unless β is large, less temporal redundancy can be removed by interframe and interpolative coding this way, and compression becomes less efficient. A judicious choice of β is important because large β also

means lesser degree of aggregation, and hence potentially lesser degree of bandwidth sharing among different video streams.

The feedback-reference VAS sends feedback information to the MPEG sources as to which codewords have been chosen for delivery during aggregation, so that their respective encoders can put all delivered components into their Frame Storages. Since the delivery of all forwarded data in the aggregated stream is guaranteed by the network, error propagation will not occur. Compared with the partial-reference VAS, the feedback mechanism here increases the encoders' compression efficiency.

To further illustrate the subtlety of feedback-reference VAS and its advantage over partial-reference VAS, let us consider three blocks in three successive frames. Suppose that all signals in block a (in frame 1) have been transmitted. Further suppose that block b (in frame 2) is interframe coded with motion compensation based on block a and that only the lower half (in frequency domain) of the ME errors in block b are chosen during aggregation. In response to feedback from the feedback-reference VAS server, the MPEG coder puts back only the lower half of that ME errors into the Frame Storage. In other words, in the Frame Storage, only the lower half signals are updated to correspond to signals in block b, while the higher half still correspond to signals in block a. As a result, when block c (in frame 3) is interframe coded, not only redundancy in the lower half signals can be removed based on block b, redundancy in the higher half signals can also be removed based on block a. A disadvantage of the feedback-reference VAS is that real-time control of the MPEG encoders is required, which is cumbersome under certain situations. For instance, when the video sources are pre-compressed and stored in the disks for future display,

this VAS requires decoding and then re-coding of the video sequences during the aggregation process. With the partial-reference VAS, on the other hand, decoding and re-coding are not necessary since the pre-compressed stored video could be coded in a compatible way such that only the first β codewords of each block are put in the Frame Storages as references.

Avoidance of error propagation in the above two classes of VAS reduces compression efficiency. A full-reference VAS simply ignores, rather than avoids, error propagation. Thus, at the encoders, all data of reference frames will be put into the Frame Storages (regardless of whether they will be transported). At the receiver side, all the received data of reference frames will be stored at the decoder's Frame Storage. In general, for a given bandwidth, more higher-frequency components can be sent with this approach as more redundancy can be removed. However, the received signals may contain propagated errors due to discrepancies of the data in the sender's and receiver's Frame Storages.

It is difficult to compare the full-reference and partial-reference VAS's from the viewpoint of image quality, as this involves the comparison between degradation due to error propagation and less efficient compression, which depends to a large extent on the scene contents. Nevertheless, when the texture complexity of a video sequences is rather steady (e.g., in video-conferencing), we expect the full-reference VAS to provide better image quality. This is because when successive frames are strongly correlated, the ME error and hence degradation due to error propagation is small. By the same token, the partial-reference VAS should be better when successive frames are not strongly correlated (e.g., video with fast motions). Because a full-reference VAS requires no modification at all on the the standard MPEG encoder, we used it for our experiments described

in the next section.

3.4 Experimental Results

This section presents experimental results that show the performance of video aggregation as compared to cell-level multiplexing with the two-layer approach. In addition, effects of varying the amount of allocated bandwidth and the number of aggregated streams on the performance of video aggregation are also studied.

The video sequences used in the experiments are 8 seconds in duration. The resolution and frame rate are 320×240 and 30 frames per second, respectively (i.e., quarter size of the NTSC standard). All of them were captured from unrelated scenes in the movie *Jurassic Park*, and were coded by an MPEG encoder with $N = 10, M = 3$ (see Section 2.1). The traffic in terms of bits per frame of one of the sequences is shown in Fig. 3.3. Note that sharp peaks occur periodically because of intraframe coding of MPEG coding scheme. In addition, the local average rate of the traffic (say, averaged over 30 frames) varies over time, thanks to the changes of scene complexity. Some traffic statistics of all the sequences are tabulated in Table 3.1. As can be seen, the traffic of all the sequences is rather bursty.

For both the aggregation and cell-level multiplexing experiments, only the transmission of the header information, MV's and DC components was guaranteed (i.e., $\beta = 1$), while the AC codewords could be discarded when the allocated bits were not enough to accommodate all data. For aggregation, the metric used for comparing the image qualities among the MB's was noise energy (see Section 3.2.2).

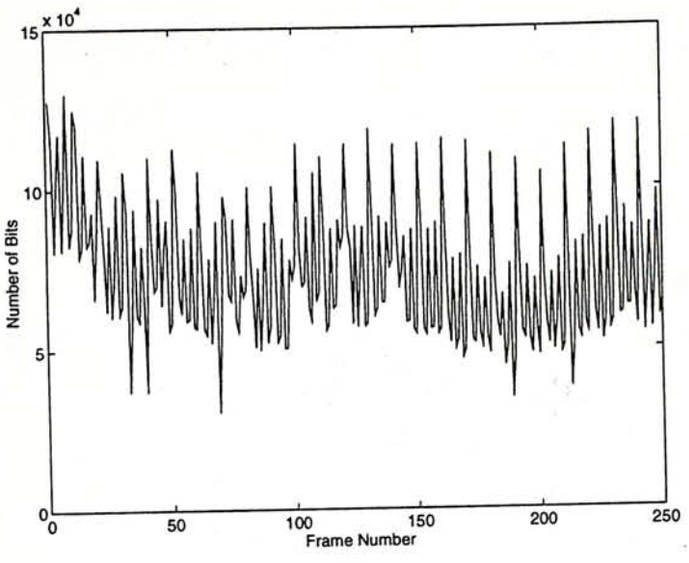


Figure 3.3: Bits per frame for the sequence JP4.

Table 3.1: Bits per frame of the sequences (in kbits)

Sequence Name	*Bits per Frame*	
	Mean	Standard Deviation
JP1	131.7	28.0
JP2	35.3	20.3
JP3	61.4	32.9
JP4	74.0	20.8
JP5	142.1	31.6
JP6	112.9	31.6
JP7	92.2	22.8
JP8	106.3	24.3

Both aggregation and cell-level multiplexing operations were slotted into slice periods.[2] In each slice period, a fixed number of bits corresponding to the reserved CBR bandwidth were allocated. As a simple means to reduce burstiness of the traffic to be aggregated/multiplexed, the I frames of the sessions were disaligned: the first sequence started with frame 1 (the I frame), the second sequence with frame 2, and so on. As a preliminary study, for aggregation, the effect of error propagation was simply ignored (i.e., full-reference VAS was used).

In Subsection 3.4.1, we compare the performance of video aggregation with cell-level multiplexing from the viewpoint of image quality for a given bandwidth. The effects of varying the amount of bandwidth reserved and the number of video sequences are discussed in Subsections 3.4.2 and 3.4.3, respectively

3.4.1 Comparison of Video Aggregation and Cell-level Multiplexing

Eight video sequences were used in this set of experiments. The reserved bandwidth of the CBR channel was fixed to be the sum of the mean bit rates of the sequences, where the mean bit rate of a sequence was obtained by averaging over all frames within the eight-second duration of the sequence. For simplicity, we assumed that all the 48-byte payload of the ATM cells could be used to carry data from the video streams (i.e., unlike in [10], we assumed no additional header overhead was introduced during packing of data into cells). For our video sequences, the sum of the mean rates corresponds to 132 cells per slice period. With such bandwidth usage, the average percentage of data lost in aggregation

[2]Buffers can be used in both aggregation and multiplexing systems to store excess data so that they can be transmitted in the next slice period (i.e., to allow bandwidth sharing across successive slice periods). In principle, this should provide better performance.

and cell-level multiplexing are 6.19% and 7.70%, respectively. The latter has a higher percentage loss because some bandwidth is wasted due to layering of data: as each cell can contain data from one layer only, for each sequence, there can be two non-fully packed cells (one from each layer) during a slice period.

1. Smoothness of Quality within a Frame

The original and the reconstructed images after multiplexing and aggregation for a randomly chosen frame [3] are shown in Fig. 3.4. Compared with the original image (Fig. 3.4 (a)), the post-aggregation image (Fig. 3.4 (c)) is a little "misty", as some of the high frequency signals have been discarded. Note that, however, the quality is smooth within the whole frame. For the post-multiplexing image (Fig. 3.4 (b)), although the left side is very well reconstructed, serious degradation and blocky effects can be easily seen on the right.

For a frame, let us define the SNR of an MB j (expressed in dB) as

$$SNR_{MB_j} = 10 \log_{10} \frac{\sum_l s_l^2}{\sum_l (s_l - s_l')^2} \tag{3.2}$$

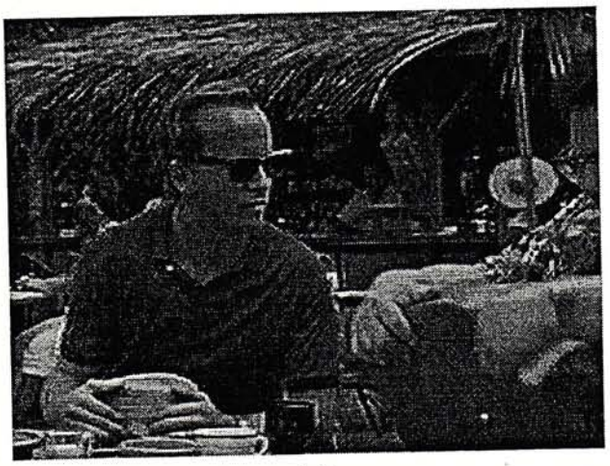
where s_l is the original (pre-MPEG compressed) value of pixel l, s_l' is the pixel value after aggregation or multiplexing, and the summations are taken over all pixels l in the MB. The smoothness of the image quality of a frame can be measured objectively by the standard deviation of SNR_{MB_j} over all MB's in the frame,

$$\sigma_{MB} = \sqrt{\frac{\sum_j (SNR_{MB_j} - \overline{SNR_{MB}})^2}{\text{Number of MB's in frame}}} \tag{3.3}$$

[3] The MPEG-coded video files for all the 8 video sequences before and after aggregation/multiplexing can be obtained from FTP site "ftp.erg.cuhk.hk" under the directory "/pub/paper/ie/broadband/vas".

(a)

(b)



(c)

Figure 3.4: A frame in the sequence JP1 (a) before MPEG coding, (b) reconstructed from the cell-level multiplexing scenario, and (c) reconstructed from the aggregation scenario.

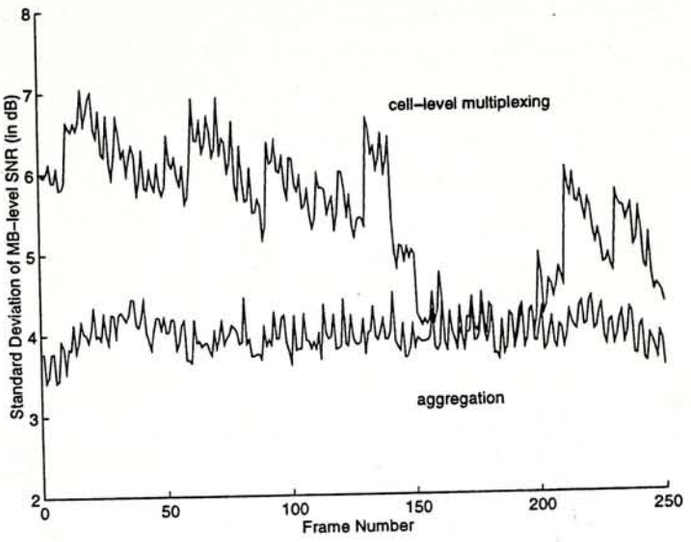


Figure 3.5: Standard deviation of SNR of the MB's in a frame, σ_{MB}, along the sequence $JP1$.

where the summation is over all MB's in the frame and

$$\overline{SNR_{MB}} = \frac{\sum_j SNR_{MB_j}}{\text{Number of MB's in frame}} \tag{3.4}$$

The larger the σ_{MB} of a frame, the less smooth is the image of the frame. Figure 3.5 plots σ_{MB} along one of the sequences after aggregation/multiplexing. As can be seen, the aggregated sequence has much lower σ_{MB} for most of the frames.

Thus, both subjectively and objectively, we have shown that aggregation provides much smoother quality within a frame than cell-level multiplexing does.

2. Quality Degradation due to Aggregation and Multiplexing

Subjectively, we can see from Fig. 3.4 that the quality of the post-aggregation image is superior to that of the post-multiplexing image.

For objective comparison among frames, frame-level SNR defined as follows

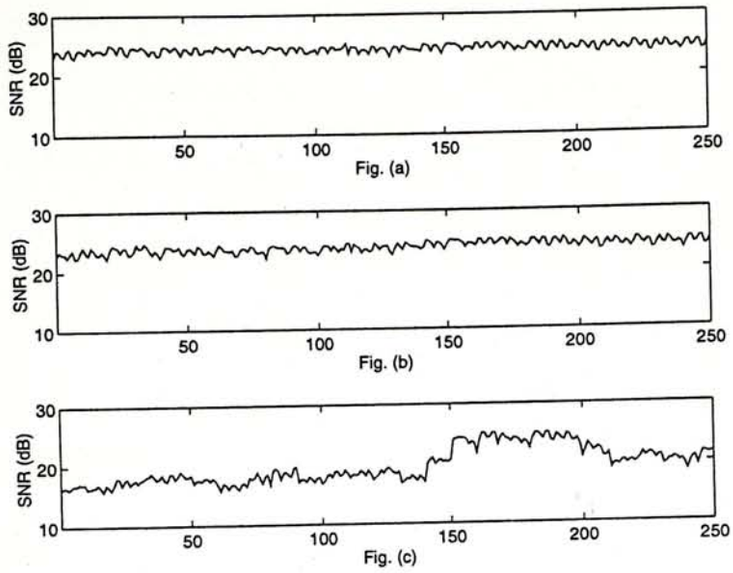


Figure 3.6: SNR of the frames in the sequence $JP1$ (a) before aggregation/multiplexing, (b) after aggregation and (c) after multiplexing.

is used to measure the image quality of a frame:

$$SNR = 10 \log_{10} \frac{\sum_l s_l^2}{\sum_l (s_l - s_l')^2} \tag{3.5}$$

which is similar to (3.2) except that the summations are over all pixels in the frame.

The SNR for all frames along the sequence $JP1$ before and after aggregation/multiplexing are plotted in Fig. 3.6. Comparing the post-aggregation sequence (Fig. 3.6(b)) with the post-multiplexing one (Fig. 3.6(c)), the former has higher and more steady SNR.

The post-aggregation sequence can have more steady image quality because even when the allocated bandwidth is not sufficient to accommodate all the data from all the eight video sequences (e.g., the first 100 frames in Fig. 3.7), the image-quality degradation of the frames is minimized by dropping the least important data. As a result, the image quality of these frames is not much different from the perfect quality received when the bit rate of the total traffic

is lower than the allocated bandwidth (e.g., frames 150 to 200 in Fig. 3.7). In contrast, with cell-level multiplexing, when the allocated bandwidth is insufficient, some important data in the ES cells may be dropped, resulting in more serious degradation.

In our experiment, the SNR results of all the eight sequences are qualitatively similar to that of the $JP1$ shown here. It does not yield additional insight and information to present all these results. In the following, we present some processed SNR statistics (i.e, mean and standard deviation) of these sequences.

To focus the image quality *degradation* resulting from aggregation/multiplexing, let us define the SNR difference of a frame, ΔSNR, to be the difference between the SNR immediately before and after aggregation/multiplexing. Therefore, higher ΔSNR means more signal energy has been dropped during aggregation/multiplexing, and therefore more serious degradation in image quality.

For notational clarity, let us use index i to refer to a frame position and index k to refer to the sequence. Thus, SNR_{ik} is the SNR of frame i of sequence k, and ΔSNR_{ik} is the SNR difference of frame i of sequence k.

For each of the sequence k, let us define ΔSNR_k and $\sigma_{\Delta SNR_k}$ to be the mean and standard deviation, respectively, of the SNR difference over all frames in the sequence:

$$\Delta SNR_k = \frac{\sum_i \Delta SNR_{ik}}{\text{Number of frames in sequence } k} \tag{3.6}$$

and

$$\sigma_{\Delta SNR_k} = \sqrt{\frac{\sum_i (\Delta SNR_{ik} - \Delta SNR_k)^2}{\text{Number of frames in sequence } k}} \tag{3.7}$$

We use ΔSNR_k as the metric for measuring the average image quality of sequence k, while $\sigma_{\Delta SNR_k}$ for the steadiness of the image quality of the sequence.

Table 3.2: Mean SNR differences, ΔSNR_k, of the sequences (in dB)

Sequence Name	ΔSNR_k	
	Aggregation	Multiplexing
JP1	0.40	4.64
JP2	0.42	0.09
JP3	0.56	3.06
JP4	0.68	3.16
JP5	0.43	5.15
JP6	0.35	2.18
JP7	0.39	1.92
JP8	0.46	2.45
Mean (ΔSNR)	0.46	2.83

The ΔSNR_k and $\sigma_{\Delta SNR_k}$ for aggregation and multiplexing for the eight sequences are given in Table 3.2 and 3.3, respectively. We see that video aggregation provides better and more steady image quality than multiplexing does for all but one sequence.

In order to compare the overall image quality of the sequences provided by aggregation and multiplexing, we further look into the mean of ΔSNR_k and $\sigma_{\Delta SNR_k}$ across the eight sequences, defined as

$$\overline{\Delta SNR} = \frac{\sum_k \Delta SNR_k}{n} \tag{3.8}$$

and

$$\overline{\sigma_{\Delta SNR}} = \frac{\sum_k \sigma_{\Delta SNR_k}}{n} \tag{3.9}$$

where n is the number of sequences being aggregated/multiplexed (i.e., $n = 8$ in this experiment). As both $\overline{\Delta SNR}$ and $\overline{\sigma_{\Delta SNR}}$ are lower in the aggregation scenario, we conclude that overall, aggregation can achieve better and more steady image quality for the sequences than multiplexing can.

Table 3.3: Standard deviation of SNR differences, $\sigma_{\Delta SNR_k}$, of the sequences (in dB)

Sequence Name	$\sigma_{\Delta SNR_k}$	
	Aggregation	Multiplexing
JP1	0.38	2.53
JP2	0.38	0.27
JP3	0.54	3.49
JP4	0.55	1.72
JP5	0.40	2.61
JP6	0.41	2.18
JP7	0.38	2.04
JP8	0.45	2.12
Mean ($\overline{\sigma_{\Delta SNR}}$)	0.44	2.12

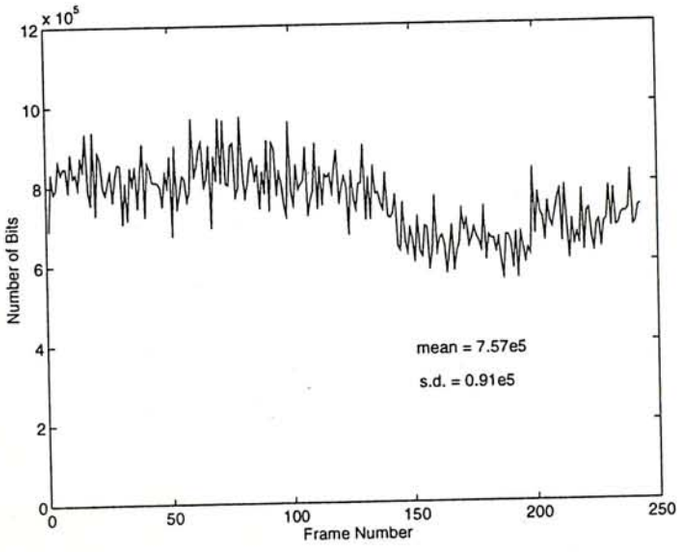


Figure 3.7: Bits per frames for the total traffic of $n = 8$

3. Fairness among the Sequences

As a metric for evaluating the fairness of image quality across the n sequences, let us define $\sigma_{\Delta SNR_i}$ to be the standard deviation of ΔSNR_{ik} over the n sequences in frame period i:

$$\sigma_{\Delta SNR_i} = \sqrt{\frac{\sum_k (\Delta SNR_{ik} - \Delta SNR_i)^2}{n}} \tag{3.10}$$

where

$$\Delta SNR_i = \frac{\sum_k \Delta SNR_{ik}}{n} \tag{3.11}$$

is the mean SNR difference across the n sequences in the frame period i. Larger $\sigma_{\Delta SNR_i}$ means that in frame period i, the image-quality degradation of the sequences is less uniform; in other words, there is a lower degree of fairness among n the sequences.

$\sigma_{\Delta SNR_i}$ for all frame periods for both aggregation and multiplexing scenarios are plotted in Fig. 3.8. Note that for all frame periods, $\sigma_{\Delta SNR_i}$ is much lower in the aggregation scenario. This verifies that aggregation can achieve better fairness among the sequences than multiplexing can.

3.4.2 Varying Amount of the Allocated Bandwidth

Table 3.4 summarizes the results when the bandwidth allocated (normalized by the mean rate, i.e., 132 cells per slice period) to the eight sequences being aggregated or multiplexed is varied. Specifically, $\overline{\Delta SNR}$ and the corresponding percentage of data lost are given.

As expected, for both aggregation and cell-level multiplexing, when less

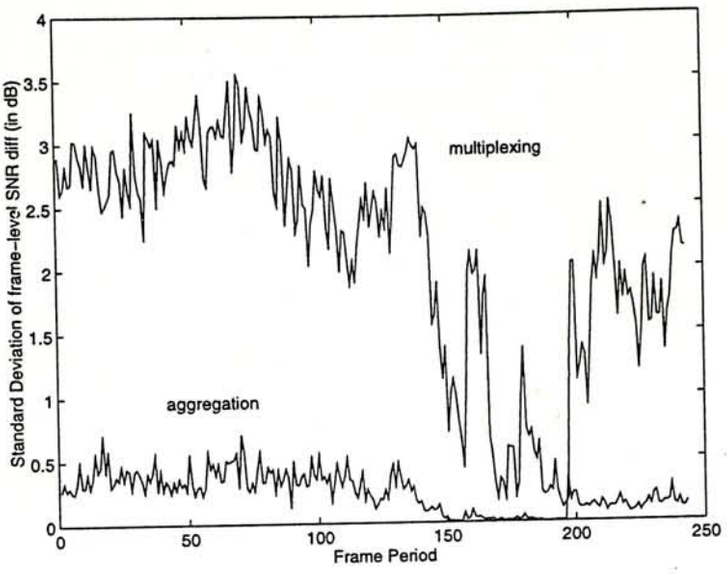


Figure 3.8: Standard deviation of SNR difference across the sequences, $\sigma_{\Delta SNR_i}$ (in dB)

Table 3.4: Mean SNR difference across all frames and across the 8 sequences, $\overline{\Delta SNR}$ (in dB) and percentage of data loss for different amount of bandwidth allocated (normalized by the mean rate).

Allocated Bandwidth	$\overline{\Delta SNR}$		*Mean Percentage of Data Loss*	
	Aggregation	Multiplexing	Aggregation	Multiplexing
1.0	0.46	2.83	6.19	7.70
0.9	0.86	4.28	11.67	13.69
0.8	1.48	5.75	19.94	21.48
0.75	1.79	6.49	23.93	26.13
0.6	3.10	8.26	38.96	40.22
0.5	4.04	9.10	47.85	49.70

Table 3.5: Image quality of the sequences transmitted by video aggregation with different n

n	$\overline{\Delta SNR}$ (in dB)	$\overline{\sigma_{\Delta SNR}}$ (in dB)
8	0.46	0.44
4	0.89	0.73
2	3.96	1.32
1	6.16	1.54

bandwidth is allocated, more data is dropped and hence $\overline{\Delta SNR}$ increases. Nevertheless, for the same bandwidth allocation, aggregation always provides better image quality than cell-level multiplexing does. Alternatively, for the same image-quality requirement, aggregation uses less bandwidth. Note that if the tolerable image degradation is up to 2.8 dB, the reduction of bandwidth usage by aggregation is more than 25% with respect to the mean bandwidth of the MPEG video streams, which is needed by cell-level multiplexing with two-layer coding.

3.4.3 Varying Number of Sequences

We now present results related to varying the number of video sequences, n, in video aggregation. In all cases, the allocated bandwidth of the CBR channel is equal to the sum of the mean rates of the n sequences.

Table 3.5 gives both $\overline{\Delta SNR}$ and $\overline{\sigma_{\Delta SNR}}$ (defined in (3.8) and (3.9), respectively) over all the n sequences as n is varied.

As n is reduced, both $\overline{\Delta SNR}$ and $\overline{\sigma_{\Delta SNR}}$ increase. In other words, when fewer sequences are aggregated, the average degradation and steadiness of image quality also decrease. This is because when n is reduced, the total input

traffic becomes more bursty. Compare, for example, Fig. 3.7, where $n = 8$, with Fig. 3.9, where $n = 2$. When n is small, sharp peaks occur whenever one of the sequences outputs at high bit rate (e.g., due to I frames or scene changes). At other times, the total bit rate remains lower than the mean rate. As long as aggregation is slotted into slice periods, data will be dropped when the peaks occur. Meanwhile, all data can be transmitted (even with unused excess bandwidth) at other times. As a result, the degradation becomes severe at peak times and the image quality is not steady over time. When n is large, the sharp peak of a sequence can be absorbed by a larger number of other sequences that do not need that much bandwidth at that moment in time. Consequently, the degradation is less severe and the image quality more steady.

The above observation suggests that when n is small (e.g., $n < 4$), it is better to have temporal statistical multiplexing in addition to the bandwidth sharing among the sequences. As mentioned before, temporal statistical multiplexing can be achieved by having a smoothing buffer at the output of the VAS server (the flow control block in Fig. 3.2). Bandwidth allocated to each slice period is time varying and it depends on two factors: the current buffer-occupancy and the the intrinsic bandwidth demand for the slices being aggregated, which could be measured using the bit-versus-distortion function employed in the aggregation process. The exact algorithm for bandwidth allocation within this framework remains a subject of further study. Although better image quality can be obtained with temporal smoothing, more complicated operation in the VAS server will be needed. Furthermore, the delay and delay jitter introduced at the buffer must also be dealt with at the receiving end. Fortunately, when n

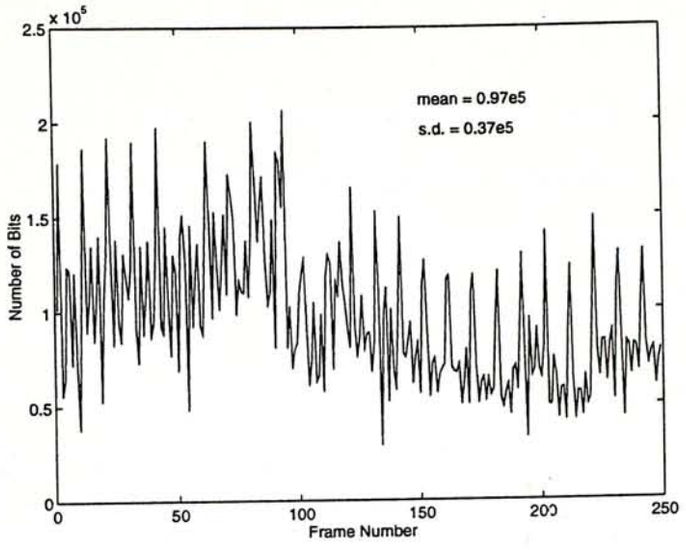


Figure 3.9: Bits per frames for the total traffic of $n = 2$

(say, $n \geq 8$) is sufficiently large, smoothing the input traffic solely in the spatial domain (across the sequences) with aggregation does provide good-quality image, obviating this complicated operation.

For cell-level multiplexing, as we have already seen, smoothing in the spatial domain is not enough even when n is as large as 8. Thus, we expect that temporal smoothing with a buffer will have a more beneficial effect here. Nevertheless, as indicated in Fig. 3.7, besides the bit rate variations from frame to frame, the average bit rate (say averaged over 30 frames) also varies on a larger time scale. This can not be smoothed out with a buffer size that is reasonably small (say smaller than 30 frames) to avoid large delay and delay jitter. Therefore, we would still expect data to be dropped during the peaks of the average bit rate. When this occurs, aggregation will prove to be superior to cell-level multiplexing.

3.5 Conclusion

This chapter has investigated video aggregation, a concept that integrates compression and multiplexing of video information. In video aggregation, a bulk of fixed bandwidth is allocated to a group of video sessions, and it is up to the video sessions to adapt their traffic to the fixed bandwidth. With the fixed CBR output, video aggregation frees the network operator from the complicated bandwidth-allocation and tariff problems. It has been shown experimentally (based on the objective SNR measure and subjective observation of image quality) that video aggregation can provide better image quality than multiplexing at the cell level. In particular, two important goals are achieved: 1) smooth and good image quality for the frames of each video session, and 2) fairness of image quality among the video sessions.

3.6 Appendix: Alternative Implementation of MPEG Video Aggregation

This appendix describes two alternative approaches for implementation of MPEG video aggregation. For convenience, we shall assume that the aggregation is slotted into frame period (as opposed to slice period in the experiments described in Section 3.2.2). Note that in general, the unit of aggregation can be smaller or larger, depending on the processing capability.

3.6.1 Profile Approach

For an arbitrary frame j, we define the "profile", $R_j(D)$, as the number of bits needed to achieve a distortion of D. Section 4.5 will describe in details how we can obtain the profile of a frame in MPEG-coded video sequences.

After obtaining the profiles of each of the n frames being aggregated, the operating distortion level D' is given by (see Section 3.1.1)

$$R_1(D') + R_2(D') + \cdots + R_n(D') = B_t \tag{3.12}$$

where B_t is the reserved bandwidth for each frame period. The n frames are then further compressed at the distortion level D'.

Comparing with the implementation described in Section 3.2.2, this profile approach can avoid the sorting process which is needed for identifying the MB which currently has the highest distortion, and hence can help reducing the computational complexity of the video aggregation process.

3.6.2 Bit-Plane Approach

Both the implementation described in Section 3.2.2 and the profile approach are in "frequency-plane" (see Section 2.3), in the sense that if the bandwidth is not sufficient to accommodate all the data, data will be dropped from the higher frequencies. However, we can also implement MPEG-video aggregation in the "bit-plane" , thus, drop data from the least significant bits (see Section 2.3).

For MPEG video, many researchers [12, 13] related the quantization factor Q_j used for coding a frame j, with the corresponding output bit count by

$$X_j = S_j \times Q_j \tag{3.13}$$

where X_j is called the "global frame complexity " of the frame, and S_j is the number of bits generated by coding it. As long as this relationship holds, for each frame j, the numbers of output bits with different Q_j used can be predicted.

Since the quantization distortion of a frame is given by $Q^2/12$, [4] video aggregation for n frames can be achieved by coding all those frames with a common quantization factor Q', such that

$$\frac{X_1}{Q'} + \frac{X_2}{Q'} + \cdots + \frac{X_n}{Q'} = B_t \tag{3.14}$$

Comparing the frequency-plane approaches described in Sections 3.2.2 and 3.6.1, this bit-plane approach needs not calculate the signal energy for every codeword, however, re-quantization of the frames requires additional quantization and VLC coding. Note that it is difficult to ensure the total output bit count in each frame period (from coding the n frames) to be exactly B_t, because 1) the relationship in (3.13) is only an approximate estimation, and 2) for successive choices of Q, the total output bit counts may deviate much. Therefore, if we want to smooth the traffic of a bundle of MPEG video to CBR, MPEG video aggregation implemented with this bit-plane approach must be followed by temporal smoothing.

[4] This can be derived based on the assumption that the amplitudes of the DCT coefficients are evenly distributed within each of the quantization steps.

Bibliography

[1] D. Deloddere, W. Verbiest and H. Verhille, "Interactive Video On Demand", *IEEE Comm. Magazine*, May 1994.

[2] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta and J. Hill, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks", *Proc. IEEE Infocom 93*, pp. 919–926.

[3] S. S. Dixit and P. Skelly, "Video Traffic Smoothing and ATM Multiplexer Performance", *Proc. IEEE Globecom 91*, pp. 0239–0243.

[4] R. Coellco and S. Tohme, "Video Coding Mechanism to Predict Video Traffic in ATM Network", *IEEE Globecom 93*, pp. 447–450.

[5] D. M. Cohen and D. P. Heyman, "Performance Modeling of Video Teleconferencing in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 6, December 1993.

[6] N. M. Marafih, Y-Q. Zhang and R. L. Pickholtz, "Modeling and Queuing Analysis of Variable-Bit-Rate Coded Video Sources in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 4, No. 2, April 1994.

[7] G. Ramamurthy and B. Sengupta, "Modeling and Analysis of a Variable Bit Rate Video Multiplexer", *Proc. IEEE Infocom 1992.*

[8] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 3, June 1993.

[9] M. De Prycker, *Asynchronous Transfer Mode : Solution for Broadband ISDN*, Ellis Horwood, 1993.

[10] M. Ghanbari and C. J. Hughes, "Packing Coded Video Signals into ATM Cells", *IEEE/ACM Trans. on Networking*, Vol. 1, No. 5, October 1993.

[11] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Commun. Magazine*, May 1994.

[12] L. Wang, "Bit Rate Control for Hybrid DPCM/DCT Video Codec", *IEEE Trans. Circuit and Systems for Video Tech.*, Vol. 4, No. 5, October 1994.

[13] L. W. Lee, J. F. Wang, J. Y. Lee and C. C. Chen, "On the Error Distribution and Scene Change for the Bit Rate Control of MPEG", *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 3, August 1993.

Chapter 4

A Control-Theoretic Study of Video Traffic Adaptation

To facilitate temporal smoothing of video traffic, data from the video encoder is fed to a smoothing buffer, which then forwards data at a constant rate to the network. In order to avoid cell loss due to overflow at the smoothing buffer, which may result in serious image-quality degradation, the video encoder should *adapt* its output traffic in response to the buffer occupancy. When the buffer occupancy is relatively high, the encoder should reduce its output bit rate by coding the video sequence at lower image-quality. On the other hand, when the buffer occupancy is low, the video will be coded at better image-quality. In this chapter, we simply refer this close-loop approach for temporal smoothing by *adaptation* unless specified.

As discussed in Section 1.1, fluctuating bit rate is inherent to VBR video coding, because of 1) switching of coding modes, and 2) variations of scene contents of the underlying video. This makes the objectives of video traffic

adaptation schemes different from those for conventional flow-control schemes for data traffic, in the sense that it should keep the the image-quality for the frames steady, while buffer-occupancy fluctuations within a reasonable region is expected.

Many previous researchers have investigated adaptation schemes for VBR-coded video traffic [1, 2, 3, 4]. However, as we will see, their schemes are somewhat *ad hoc* in nature, and are not analyzable.

This chapter, in contrast, studies the generic video traffic adaptation scheme with a linear-feedback model, which can be analyzed systemically. As a result, we can, 1) design and fine-tune the adaptation scheme, and 2) study the trade-off between steadiness of image-quality and buffer-occupancy fluctuations (this shall be discussed in Section 4.1.1) in a systematic and quantifiable manner. While we believe that the particular linear-control model proposed in this chapter would perform well for most video sequences, perhaps the more important contribution is that this framework enables analytical study of video traffic adaptation scheme. This in turn may open up many other possibilities for future investigation.

The remainder of this chapter is organized as follows. Section 4.1 reviews some previous adaptation schemes for VBR video encoders with a generic model. Section 4.2 gives motivations for studying video traffic adaptation in a control-theoretic approach. Section 4.3 describes an analytical model for the adaptation process. Section 4.4 analyzes the stability and operation of the adaptation model. Section 4.5 describes the implementation of our adaptation scheme on MPEG video sequences. Experimental results of transmitting an MPEG video sequence with our adaptation scheme are given in Section 4.6. The conclusion

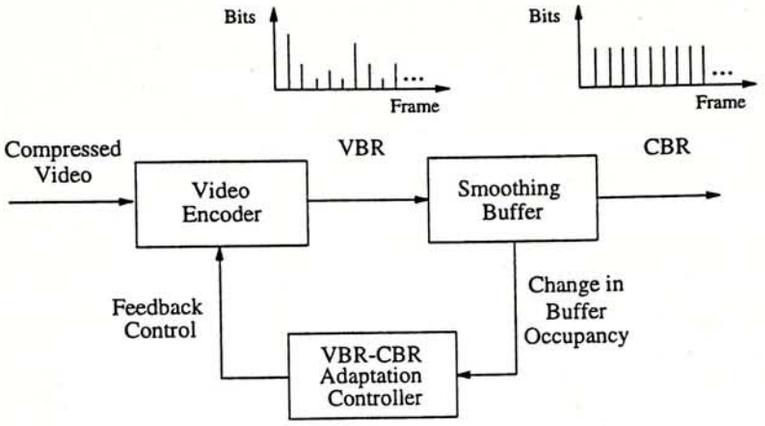


Figure 4.1: A generic model for video traffic adaptation which employs temporal smoothing.

of this chapter is in Section 4.7.

4.1 Review of Previous Adaptation Schemes

Detailed review for some previous adaptation schemes for MPEG-like video encoders (i.e., VBR video encoders which employ several coding modes) are in Appendix II (Section 4.9) [1, 2, 3, 4]. In this section, we try to summarize and evaluate them with respect to a generic model.

4.1.1 A Generic Model for Adaptation Scheme

Figure 4.1 shows a generic model for video traffic adaptation. There are three components in the system: the video encoder, the smoothing buffer, and the VBR-CBR adaptation controller.

In each time interval Δt (e.g., for MPEG coding, Δt can be a MB's period [1], a frame period, or even a SGOP period [2]), a constant number of bits are

removed from the buffer for transmission on its output, and a random number of bits are produced by the encoder and put into the buffer. The difference in input and output is revealed in the change in buffer occupancy, which is conveyed to the adaptation controller. Processing on this information, the adaptation controller in turn decides how to control the encoder via a feedback path. Majority of previous adaptation schemes specify the control message in terms of change in the quantization factor [1, 2, 3, 4] , ΔQ, used in the compression process (see Section 2.1). In this chapter, however, we assume that the feedback control message is specified in terms of a change in bit rate, Δr. This is a more general model: we can map Δr into ΔQ if needed.

Particularly for the proposed adaptation schemes for MPEG-like encoders described in Appendix II, the control message is determined based on the difference between the coded and target bit counts in the preceding interval, while the target bit count for each interval is allocated dynamically in a way to try to restore the buffer occupancy (to its desired value) at the end of next H GOP's (following the current GOP). Note that H can be zero as in [1].

4.1.2 Objectives of Adaptation Controller

Central to an adaptation system is the design of the adaptation controller. When there are changes in the intrinsic bandwidth requirements of the video sequence, that results changes in buffer occupancy, the adaptation controller should be able to "instruct" the encoder to operate at a different image-quality objective.

In general, the variations of the intrinsic bandwidth requirements of a video can be classified into two types: long-term variations and short-term fluctuations. When the long-term intrinsic bandwidth of the video sequence becomes

higher than the output bit rate of the smoothing buffer (i.e., the reserved bandwidth of the channel), the adaptation controller has no choice but to inform the encoder to reduce its bit rate by coding the video at a lower image-quality. By the same token, when the long-term intrinsic bit requirements reduce, the controller should inform the encoder to code the video at higher image-quality so as to fully utilize the reserved bandwidth.

However, when there is short-term fluctuations of the intrinsic bit requirements of the video (e.g., due to coding-mode switching and scene change), the adaptation controller should be able to smooth out the "transient responses" in image-quality and buffer occupancy. In general, as far as the transient responses are concerned, there are three objectives for an adaptation controller:

1. minimize image-quality fluctuations,

2. minimize buffer-occupancy fluctuations, and

3. avoid unnecessary control of the encoder due to coding-mode switching.

In order to achieve smooth image-quality transitions along the frames, the control message should be determined based on both the current buffer occupancy as well as the recent trend of the variations of it. Moreover, "weak" control (i.e., with respect to the previous MPEG video adaptation schemes, large H for the target bit allocation) is preferred . This is because although the current buffer occupancy may be high, if the buffer occupancy is in a downward trend, it is not urgent to reduce the output bit rate of the encoder. Even when the current buffer occupancy is high and is going up, the traffic may smooth itself later by allowing longer period for temporal smoothing. An extreme case for this is VBR video compression (see Section 1.1) with no feedback at all. However,

to reduce buffer overflow, this requires huge buffer with output rate larger than the average rate of the video sequence. Unless peak of the video sequence is reserved for the channel, buffer overflow may still occur.

Note that to achieve steady image-quality does not necessarily require longer control interval, Δt. Although longer Δt means that more frames (being in the same interval) will be coded with the same image-quality, the accumulated change in buffer occupancy at the end of a Δt could be large. Therefore, drastic change of image-quality among different intervals may result. Nevertheless, if Δt is smaller than a frame period, as the encoder may be regulated to code different part of a frame with different image-quality, blocky effect may result.

On the other hand, to minimize the buffer-occupancy fluctuations, we may want to regulate the output bit rate of the encoder more frequently, and to restore the buffer occupancy (to its desired status) as fast as possible. This means that shorter interval Δt and "strong" control (i.e., with respect to the previous MPEG video adaptation scheme, large H for the target bit allocation) are preferred. An extreme case for this is CBR video compression (see Section 1.1).

The third objective applies to compression schemes that employ coding-mode switching. One would expect the buffer occupancy to fluctuate in accordance with the coding mode even if there were no scene change, and the scene complexity in successive frames were roughly the same. These fluctuations are "natural" and should not cause the adaptation controller to apply undue control on the encoder that might give rise to fluctuations in visual quality.

4.2 Motivation for Control-Theoretic Study

As can be seen from the above discussion, there is an engineering tradeoff in designing an adaptation controller: we cannot maintain steady image quality along the frames while we try to reduce buffer-occupancy fluctuations. To have a reasonable balance between these two contradicting properties, we need to know the effect of adjusting the parameters of the adaptation scheme on the performance of the system (i.e., image quality of the frames in the video sequence, and the buffer-occupancy fluctuations). Moreover, since different video sequences may have very different traffic characteristics (because of different scene content variations, or even structure for coding-mode switching), we should fine-tune the adaptation scheme (i.e., try to achieve a new operating point) for individual video sequence. Both of these require analysis on the operation of the adaptation scheme in a quantifiable manner. However, as the previous schemes are somewhat *ac hoc* in nature (see Appendix II), analysis can hardly be performed. This motivates us to model the generic adaptation system with an analytical framework, and to analyze it in a systematic approach.

4.3 Linear Feedback Controller Model

This section models the generic adaptation system (see Fig. 4.1) as a linear feedback control model. In the model, Δt is assumed to be a frame-period. This is because from the discussion in Section 4.1.2, a frame-period is a compromising choice for Δt. In the followings, Section 4.3.1 describes our encoder model, and Section 4.3.2 discusses the design of the adaptation controller.

4.3.1 Encoder Model

Since video traffic adaptation concerns both the image quality of the video, and the occupancy at the smoothing buffer, some knowledge about the relationship between these two factors is essential. Suppose that associated with each frame n is a bits-distortion function $f_n(D)$ that describes the number of bits required to code the frame with a distortion objective of D. This function may either be estimated or computed explicitly by the encoder. Depending on the scene contents as well as the coding modes, different frames may have different bits-distortion functions.

1. Encoder Operation

With reference to Fig. 4.2, suppose that for frame n, the number of bits output by the encoder is $R_o[n]$, and this corresponds to a distortion of $D[n]$. For frame $n+1$, the number of bits needed to maintain the same distortion (hence no image-quality fluctuations) be

$$R'_o[n+1] = R_o[n] + \Delta R[n+1] \tag{4.1}$$

where $\Delta R[n+1]$ is the bit adjustment (see Fig. 4.2).

Based on the buffer status, the adaptation controller may want to adjust the encoder output bit rate. In our model, this is done by specifying the number of bits to be reduced from $R'_o[n+1]$, $\Delta r[n+1]$ (see Fig. 4.3). The encoder then codes frames $n+1$ with the objective of outputting

$$R_o[n+1] = R'_o[n+1] - \Delta r[n+1] \quad \text{bits} \tag{4.2}$$

Note that small $\Delta r[n+1]$ means that the corresponding change in distortion

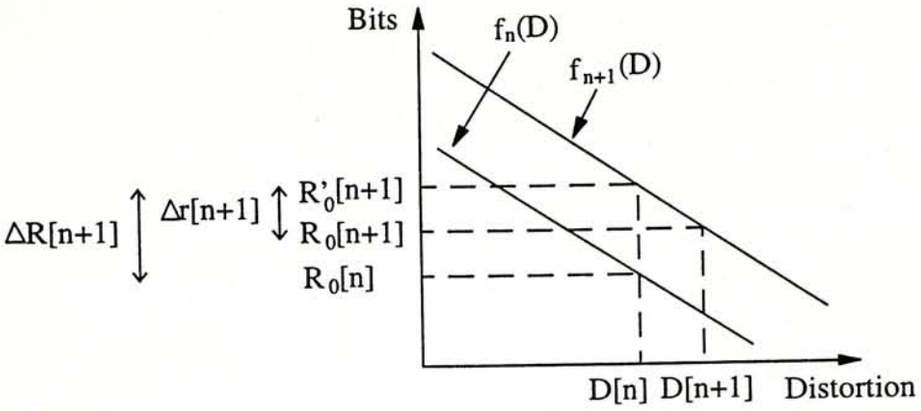


Figure 4.2: Bit adjustment at the encoder.

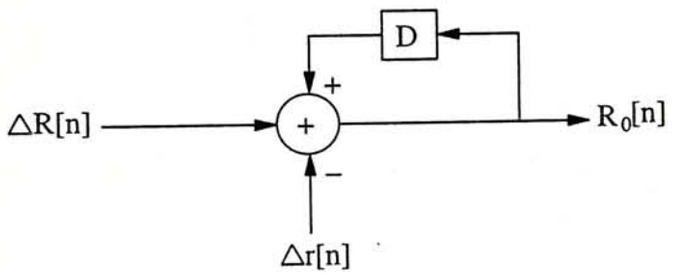


Figure 4.3: The encoder operation.

level $\Delta D[n+1] = D[n+1] - D[n]$ is also small. Therefore, the adaptation controller can minimize the image-quality fluctuations by minimizing $\Delta r[n+1]$.

Note that we have not assumed anything about the forms of $\Delta R[n]$ and $f_n(D)$ for an arbitrary frame n. The above is merely a description of the encoder operation regardless of the statistical and functional behaviour of $\Delta R[n]$ and $f_n(D)$.

2. Analytical Encoder Model

For analysis, we assume that $\Delta R[n]$, the input, is independent of the past and present feedback control $\Delta r[i], i \leq n$. It is obvious, however, that the future feedback control will be dependent on $\Delta R[n]$. The above assumption implies that for all frame n (even with different coding modes),

$$f_n(D) = -g(D) + b_n \tag{4.3}$$

where $g(D)$ is non-negative and independent of n, while b_n is independent of the past values of $\Delta r[n]$, but may vary with n. Thus, the bits-distortion functions $f_n(D)$ of all frames are of the same shape, but may have different offset b_n. Note that if $f_n(D)$ and $f_{n+1}(D)$ are of different shapes, then $\Delta R[n+1]$ will depend on $D[n]$, which in turn depends on the past values of $\Delta r[n]$. In other words, $\Delta R[n+1]$ and $\Delta r[n]$ are coupled. To illustrate this more clearly, we suppose in Fig. 4.4 that both $f_n(D)$ and $f_{n+1}(D)$ are linear, but $f_{n+1}(D)$ has a steeper slope. As can be seen, the higher the $D[n]$ is, the larger the $\Delta R[n+1]$ will be. The form of $f_n(D)$ in (4.3) will be justified for MPEG-coded video by experimental results in Section 4.5.

We define the intrinsic bandwidth $R_i[n]$ of a frame n as the number of bits required to code the frame with some fixed reference distortion D_0. We can define the distortion measure such that $D_0 = 0$. Assuming $g(D) = 0$[1], $R_i[n] = b_n$, and the encoder operation in (4.1) becomes

$$R_o'[n+1] = R_i[n+1] - g(D[n]) = R_i[n+1] - r[n] \tag{4.4}$$

[1]Note that if $g(D)$ has a constant term and $g(D_0) \neq 0$, then we may put the constant term into b_n.

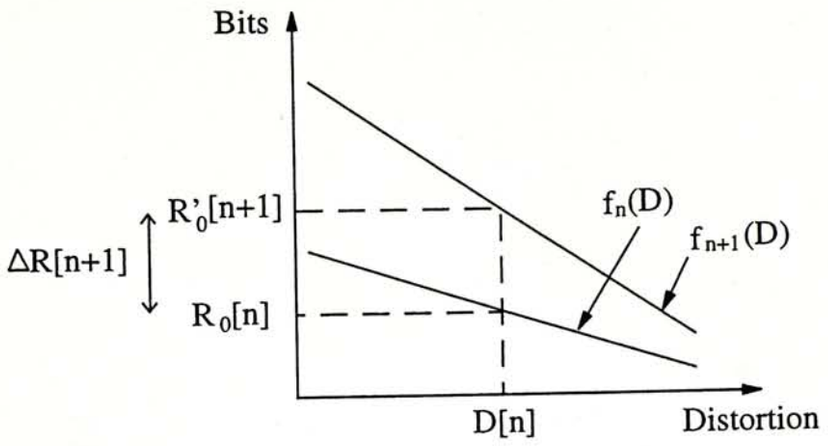


Figure 4.4: The case that $\Delta R[n+1]$ and $\Delta r[n]$ are coupled.

and the number of output bits of the encoder (in (4.2)) can be written as

$$R_o[n] = R_i[n+1] - r[n] - \Delta r[n+1] = R_i[n+1] - r[n+1] \qquad (4.5)$$

where $r[n]$ is the number of bits to be deducted from $R_i[n]$ to code frame n with distortion $D[n]$. We then arrive at the encoder model in Fig. 4.5. Note that the larger the bit reduction $r[n]$ is, the larger the distortion $D[n]$ will be. The change in the number of bits sent is

$$\Delta R_o[n+1] = R_o[n+1] - R_o[n] = (R_i[n+1] - R_i[n]) - \Delta r[n+1] \qquad (4.6)$$

Note that there are two factors influencing $\Delta R_o[n+1]$. The first is the change in the intrinsic bandwidth $(R_i[n+1] - R_i[n])$, which is not under the control of the adaptation controller. The second is the bit adjustment $\Delta r[n+1]$, which is under the control of the adaptation controller.

With the analytical encoder model in Fig. 4.5, we can study the image quality of a frame n by studying $r[n]$, and the smoothness of image-quality transition along the frames by $\Delta r[n]$.

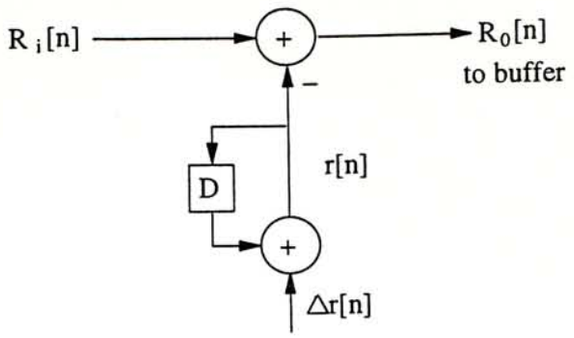


Figure 4.5: The encoder model.

4.3.2 Adaptation Controller Model

Figure 4.6 is a frequency-domain [5] block diagram that shows the encoder model being connected to the adaptation controller, which consists of two components, the buffer-status filter and the feedback compensator. In our notation, the z-transform of a time-domain signal $X[n]$ is $X(z) = \sum_{n=0}^{\infty} X[n]z^{-n}$.

We define $A[n] = R_i[n] - C$, where C is the number of bits removed from the buffer for transmission per frame period. Therefore, the change in buffer occupancy level after coding frame i is given by $\Delta B_0[n] = A[n] - r[n]$. Instead of defining $B_0[n]$ to be the buffer occupancy, we define it as the deviation from a desired buffer level, say, B, so that $B_0[n]$ can be both positive or negative. In practice, B, for instance, could be set at $B_{max}/2$, where B_{max} is the buffer size. Assuming the system starts at B, the buffer deviation is given by $B_0[n] = \sum_{i=0}^{n} \Delta B_0[i]$.

1. Buffer-Status Filter

In order to avoid unnecessary control due to coding-mode switching, instead of exercising feedback control based on $B_0[n]$, which may fluctuate naturally with

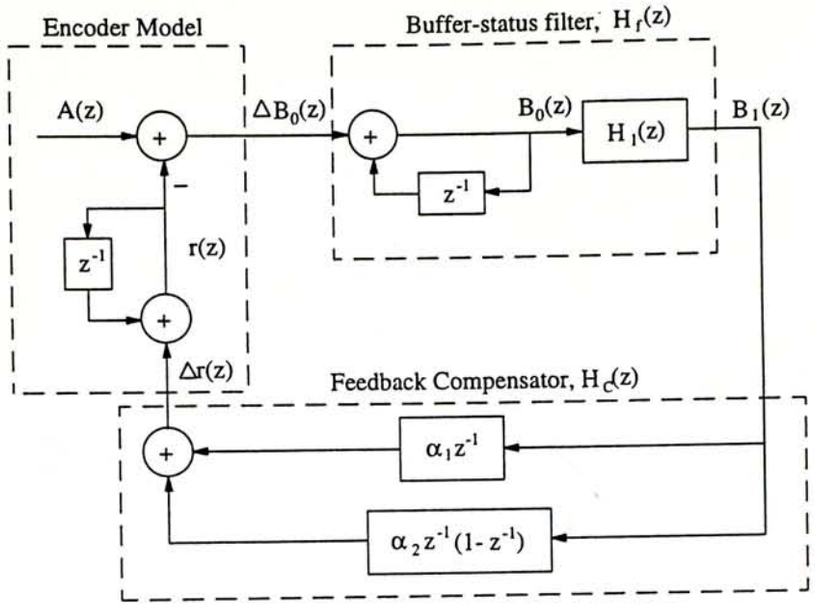


Figure 4.6: The frequency-domain block diagram of the encoder model and the adaptation controller.

the switching of coding modes, we use a filter (with transfer function $H_1(z)$) to filter out this fluctuations before the feedback is computed. Let $B_1[n]$ be the filtered response, $B_1[n]$ should be constant if $\Delta B_0[n]$ is periodic in N, the period of the coding-mode switching cycle.

A simple $H_1(z)$ that could be used is

$$H_1(z) = \frac{1}{N}(1 + z^{-1} + \cdots + z^{(N-1)}) \tag{4.7}$$

which corresponds to averaging $B_0[n]$ over the the most recent N frames (which form a complete coding-mode switching cycle). With this $H_1(z)$, the transfer function of the overall buffer-status filter is given by

$$H_f(z) = \frac{1}{N}\left(\frac{1 + z^{-1} + \cdots + z^{-(N-1)}}{1 - z^{-1}}\right) \tag{4.8}$$

2. Feedback Compensator

The feedback compensator used in this thesis is based on the PD (proportional and derivative [5]) control. Thus, the value of $\Delta r[n+1]$ that will be given to the encoder for coding the next frame is

$$\Delta r[n+1] = \alpha_1 B_1[n] + \alpha_2 (B_1[n] - B_1[n-1]) \tag{4.9}$$

where $\alpha_1, \alpha_2 \geq 0$. The associated transfer function is

$$H_c(z) = \alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1}) \tag{4.10}$$

There are two constituent compensators: the first term corresponds to DC compensation and the second term is for stabilizing the system.

Without the DC compensator, the buffer occupancy cannot be brought to the desired value B whenever there is a DC change in $A[n]$. This can be illustrated by considering a unit-step input $A[n]$. When the average buffer occupancy $B_1[n]$ becomes steady, even though it is not zero, the change of bit adjustment, $\Delta r[n]$, will be reduced to zero. As a result, $r[n]$ will keeps its value as one (i.e., $A[n] - r[n] = 0$) and therefore the backlog in the buffer will not be cleared. This will be shown analytically in the Section 4.4.3. Therefore, when $\alpha_1 = 0$, starting with the buffer level at B, a DC change will induce a change in $B_0[n]$, the deviation from B, and it is not possible to move $B_0[n]$ back to 0 if $\alpha_1 = 0$. An uncompensated DC in $B_0[n]$ is undesirable because it is then easier for the buffer to overflow (if the the DC change is positive) or underflow (if the DC change is negative) should there be any further DC changes later.

The second term in (4.9 reacts more quickly to changes in $B_1[n]$ than the first term does. An increase in $B_1[n]$ means that the buffer is filling up, and

the encoder will be requested to lower its output bit rate through this term. The goal of this term is to keep the buffer deviation small. As will be shown in Section 4.4.1, the system is stable only for a certain range of non-zero α_2.

4.4 Analysis

Instead of analyzing our adaptation system based on some stochastical model of video traffic [6, 9, 10, 11, 12] , we attempt to capture only some simple but fundamental characteristics that might be expected for general video traffic.

All the analysis in this section shall assume that the state variables are not limited by the dynamic range of the underlying physical entities. In practice, for instance, there is a limit on how large the buffer can be. However, it is a common practice in designing a feedback control system to ignore the dynamic range of the physical entities: for a well-designed and stable system, the values of the state variables are usually kept within the dynamic range anyway. In fact, rather than imposing the dynamic range as a design constraint, the dynamic range required is often determined as part of the design exercise. In the adaptation controller, for example, a goal is to minimize the buffer fluctuations, and this in turn helps us determine the buffer size required.

When we look into a video traffic $R_i[n]$, for example, the MPEG-coded video traffic shown in Figure 2.3, we can think of it as consisting of a superposition of several traffic components, as illustrated in Fig. 4.7. First, there is a "DC" component that corresponds to the scene complexity. Therefore, a DC change in $R_i[n]$ corresponds to a long-term change in the scene complexity of a video. The second component is made up of samples (impulses) here and there. Each

of this sample corresponds to a scene change that occurs at a P or B frame and makes the motion compensation not useful in reducing the bits required. The third component corresponds to coding-mode switching, and it consist of a periodic function of period N. The fourth component corresponds to small variations within the same scene. Since the feedback controller we have described in last section is a linear system, we can analyze its responses under $R_i[n]$ as a superposition of the responses under each of these components.

In the following, Section 4.4.1 studies the stability of our video traffic adaptation system. Section 4.4.2 testifies whether our system is robust against coding-mode switching. Section 4.4.3 studies the responses of a DC change by the unit-step responses, and those of a P/B-frame scene change by the unit-sample responses.

4.4.1 Stability

One goal of a feedback control system is to stabilize an unstable system. For instance, a video system consisting of an encoder and a buffer will not be stable if the average input rate to the buffer (output rate of encoder) is higher than the average output rate. The goal of the feedback is to reduce the output rate of the encoder when it is too high and increase the rate when it is too low, and hence to make sure that the best visual quality within the constraint of the communications-channel bandwidth can be achieved. It is well-known, however, that a poorly-designed feedback control system can be unstable [5]. This section investigates the range of α_1 and α_2 for stable operation of our system. Stability is defined in the *bounded-input-bounded-output* sense here: given a bounded $A[n]$, all the state variables should also be bounded.

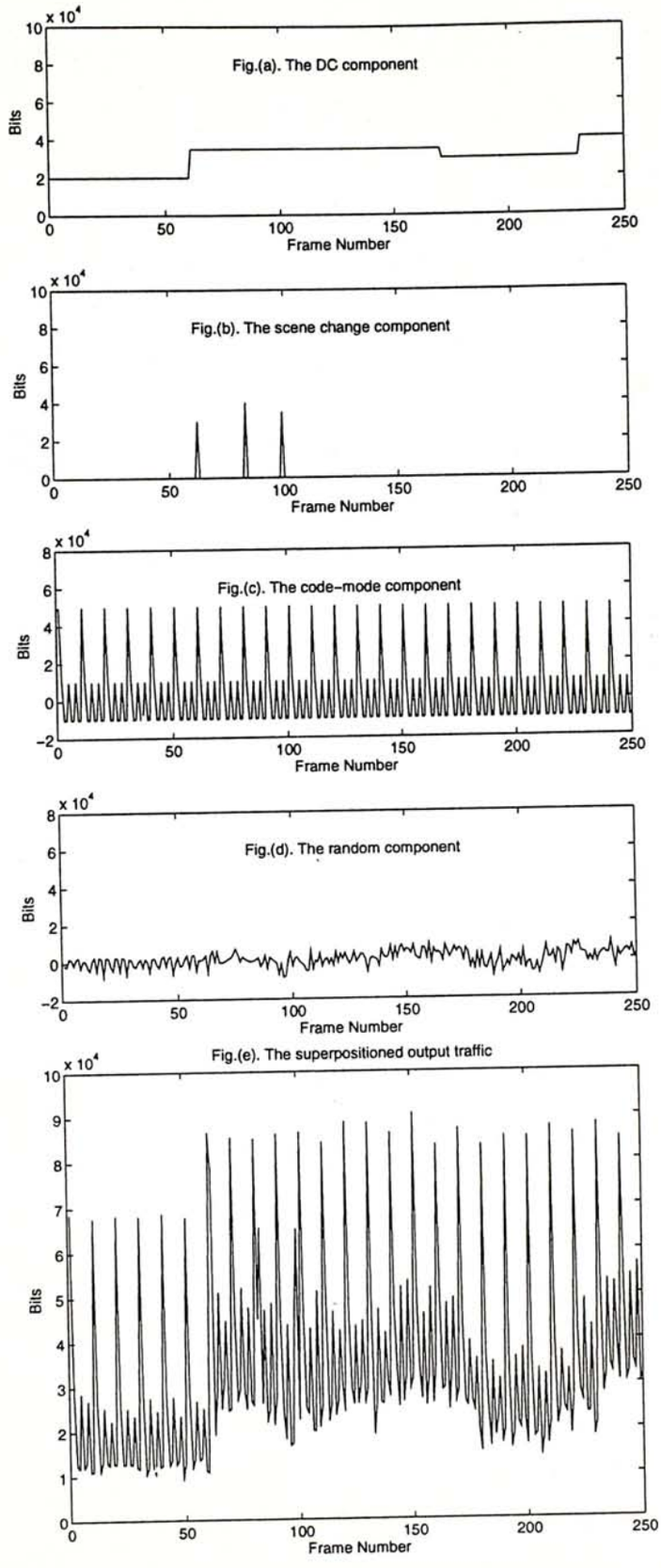


Figure 4.7: Components of an MPEG-coded video traffic

It is straightforward to derive the following transfer functions of the state variables:

$$\begin{aligned}
\frac{B_1(z)}{A(z)} &= \frac{H_f(z)}{1 + H_f(z)H_c(z)/(1-z^{-1})} \\
&= \frac{(1-z^{-N})(1-z^{-1})}{N(1-z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}(1-z^{-N})} \\
\\
\frac{B_0(z)}{A(z)} &= \frac{1}{(1-z^{-1}) + H_f(z)H_c(z)} \\
&= \frac{N(1-z^{-1})^2}{N(1-z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}(1-z^{-N})} \\
\\
\frac{\Delta r(z)}{A(z)} &= \frac{H_f(z)H_c(z)}{1 + H_f(z)H_c(z)/(1-z^{-1})} \\
&= \frac{(1-z^{-N})(1-z^{-1})\{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}}{N(1-z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}(1-z^{-N})} \\
\\
\frac{r(z)}{A(z)} &= \frac{H_f(z)H_c(z)}{(1-z^{-1}) + H_f(z)H_c(z)} \\
&= \frac{(1-z^{-N})\{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}}{N(1-z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}(1-z^{-N})}
\end{aligned} \tag{4.11}$$

Note that if $B_1[n]$ is bounded, then $B_o[n]$, $r[n]$ and $\Delta r[n]$ are also be bounded. Thus, we focus only on the transfer function of $B_1[n]$. For stability, the magnitudes of the poles of the transfer function must be smaller than one [5]. From (4.11), this means that the roots of

$$N(1-z^{-1})^2 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1-z^{-1})\}\frac{1-z^{-N}}{1-z^{-1}} = 0 \tag{4.12}$$

must have magnitudes less than one. Notice that we have divided the denominator in (4.11) by $(1-z^{-1})$ to reflect that the factors $(1-z^{-1})$ in the denominator

and numerator cancel each other.

1. Routh Test

Routh test [13] is an algebraic procedure for determining whether a polynomial has zeros in the right half-s-plane. It involves examining the signs and magnitudes of the coefficients of the equation without actually having to determine its roots. Although Routh test gives no information about the location of the roots, and hence does not indicate the relative degree of stability or instability, the test is frequently used to determine whether a system is stable.

While the detailed procedure of Routh test was described in reference [13], some special and useful results of the test can be summarized as follows [14]:

1. Necessary conditions for a polynomial to have all its roots in the left half-s-plane are:
   (a) All of the coefficients must have the same sign;
   (b) All of the powers between the highest and the lowest must have non-zero coefficients, unless all even-power or all odd-power terms are missing.
2. For quadratic polynomials, these conditions are also sufficient.
3. For a cubic polynomials, $s^3 + C_2 s^2 + C_1 s + C_0$, necessary and sufficient conditions are $C_2, C_1, C_0 > 0$ and $C_1 > C_0/C_2$.

2. Bilinear Transform

As mentioned before, the condition for stability of our discreted control system is to have all poles (i.e., roots of characteristic equation (4.12)) having magnitudes

less than one. Therefore, stability of our control system cannot be determined by means of a direct application of the Routh test. However, it is possible to apply bilinear transform [13] to the characteristic equation such that the interior (exterior) of the unit circle of the z-plane is transformed to the left (right) half of the s-plane, in other words, the condition of $|z| < 1$ can be transformed to the condition $Re[s] < 0$. Bilinear transform is defined as

$$z = \frac{s+1}{s-1} \tag{4.13}$$

thus

$$s = \frac{z+1}{z-1} \tag{4.14}$$

With this transformation, the characteristic equation (4.12) becomes

$$4N(s+1)^{N-1} + [\frac{\alpha_1}{2}(s-1)(s+1) + \alpha_2(s-1)][(s+1)^N - (s-1)^N] = 0 \tag{4.15}$$

and Routh test can be applied to determine the stability conditions for our control system.

If we express (4.15) in power series of s, we have the characteristic equation as

$$C_{N+1}s^{N+1} + C_N s^N + C_{N-1}s^{N-1} + \cdots + C_1 s + C_0 = 0 \tag{4.16}$$

where the coefficients

$$\begin{aligned} C_{N+1} &= \alpha_1 N \\ C_N &= 2N\alpha_2 \end{aligned}$$

$$C_{N-i} = \begin{cases} 4N \begin{pmatrix} N-1 \\ i-1 \end{pmatrix} + \alpha_1 \begin{pmatrix} N \\ i+2 \end{pmatrix} - [\alpha_1 + 2\alpha_2] \begin{pmatrix} N \\ i \end{pmatrix} & \text{for odd } i \\ 4N \begin{pmatrix} N-1 \\ i-1 \end{pmatrix} + 2\alpha_2 \begin{pmatrix} N \\ i+1 \end{pmatrix} & \text{for even } i \end{cases} \tag{4.17}$$

where

$$\begin{pmatrix} N \\ i \end{pmatrix} = \frac{N!}{i!\,(N-i)!} \tag{4.18}$$

3. Necessity of Buffer-Deviation Compensator

For our control system to be stable, all the roots of the transformed characteristic equation (4.16) must have their real parts as negative. According to the results of Routh test as mentioned in Section 4.4.1.1, this requires that all the coefficients of (4.16) to be non-zero and of the same sign. Look at C_N in (4.17), we can easily find that $\alpha_2 > 0$ is a necessary condition for stability.

4. Stable region for $N = 1$ and $N = 2$

For $N = 1$ and 2, the transformed characteristic equation (4.16) becomes

$$\alpha_1 s^2 + 2\alpha_2 s + (4 - \alpha_1 - 2\alpha_2) = 0 \tag{4.19}$$

and

$$2\alpha_1 s^3 + 4\alpha_2 s^2 + (8 - 2\alpha_1 - 4\alpha_2)s + 8 = 0 \tag{4.20}$$

respectively. Based on results 2 and 3 of Routh test (in Section 4.4.1.1), we conclude that the stable region (i.e., necessary and sufficient condition for the

system to be stable) for $N = 1$ is

$$4 - \alpha_1 - 2\alpha_2 > 0 \tag{4.21}$$

and that for $N = 2$ is

$$8 - 2\alpha_1 - 4\alpha_2 - 4\frac{\alpha_1}{\alpha_2} > 0 \tag{4.22}$$

5. Stable region for general N

Although Routh test can still be used for determining the stable region of a system even when the order of the corresponding characteristic equation is higher than 2, the testing procedure is tedious. Nevertheless, based on result 1 of the test, we can find a set of necessary conditions for the systems. For our control system, when $N = 10$, the necessary conditions of $C_i > 0,\ 0 \leq i \leq 11$ conform to

$$336 - 11\alpha_1 - 24\alpha_2 > 0 \tag{4.23}$$

When N is large, instead of using Routh test, we can find the boundary of the stable region of our system by studying the moving locus of the roots of the characteristic equation (4.12) when α_1 and α_2 are varied. We first study the restricted case when $\alpha_1 = 0$ before moving onto the general case $\alpha_1 \geq 0$.

5a. Restricted case: $\alpha_1 = 0$

Substituting $\alpha_1 = 0$ into (4.12), we have the following characteristic equation whose roots must be within the unit circle of the z-plane for stability:

$$N(1 - z^{-1})^2 + \alpha_2 z^{-1}(1 - z^{-N}) = 0 \tag{4.24}$$

or

$$Nz^{N-1}(z - 1)^2 + \alpha_2(z^N - 1) = 0 \tag{4.25}$$

When $\alpha_2 = 0$, there are $(N-1)$ roots at $z = 0$ and double roots at $z = 1$. Only one of the roots at 1 cancels with that of the numerator of the transfer function, and so the system is unstable.

For an infinitesimally small α_2, we show that the system is stable. We assume that the roots move smoothly when α_2 goes from zero to an infinitesimal positive number. Therefore the $(N-1)$ roots at 0 remain within the unit circle. To see what happens to the double roots at 1, substitute $z = 1+\epsilon$ into (4.25), we have

$$N(1+\epsilon)^{N-1}\epsilon^2 + \alpha_2[(1+\epsilon)^N - 1] = 0 \tag{4.26}$$

Ignore the power of ϵ higher than the second power, we have

$$\epsilon^2 + \alpha_2\{\epsilon + (N-1)\epsilon^2/2\} = 0 \tag{4.27}$$

of which roots are

$$\begin{aligned} \epsilon_1 &= 0 \\ \epsilon_2 &= -\frac{\alpha_2}{1+\frac{N-1}{2}\alpha_2} < 0 \end{aligned} \tag{4.28}$$

Therefore, one of the double roots of (4.25) remains at $z = 1$ and the other becomes smaller than one. The root at $z = 1$ cancels with that in the numerator, and so the system is stable for small α_2.

When α_2 increases to some value α_2^*, one of those roots within the unit circle moves outwards and crosses the unit circle. Therefore, α_2^* is the upper bound on α_2 for stable operation. In order to determine the value of α_2^*, we apply the transform $z = e^{j\theta}$ in (4.25), and have

$$Ne^{j(N-1)\theta}(e^{j\theta} - 1)^2 + \alpha_2(e^{jN\theta} - 1) = 0 \tag{4.29}$$

Multiply (4.29) by $e^{-j\frac{N\theta}{2}}$ and apply the relationship $(e^{j\theta} - e^{-j\theta}) = 2j\sin\theta$, we have

$$e^{j\frac{N\theta}{2}} = \alpha_2 \frac{\sin(\frac{N\theta}{2})}{2N\sin^2(\frac{\theta}{2})} j \tag{4.30}$$

Compare the modulus and the real part of both sides, we have

$$\alpha_2 = \left| \frac{2N\sin^2(\frac{\theta}{2})}{\sin(\frac{N\theta}{2})} \right| \tag{4.31}$$

and

$$\theta = \frac{(2m+1)\pi}{N}; \qquad m = 0, 1, \ldots \tag{4.32}$$

Therefore, from (4.31) and (4.32),

$$\alpha_2 = \left| 2N\sin^2\left\{ \frac{\pi}{2}\left(\frac{1+4m}{N}\right)\right\}\right| \tag{4.33}$$

The smallest value of α_2 is the critical value (the first time a root crosses the unit circle as α_2 is increased from zero) and it is obtained when $m = 0$:

$$\alpha_2^* = 2N\sin^2(\pi/2N) \tag{4.34}$$

Therefore, the stable region when $\alpha_1 = 0$ is

$$0 < \alpha_2 < 2N\sin^2(\pi/2N) \tag{4.35}$$

5b. General case: $\alpha_1 \geq 0$ and $0 < \alpha_2 < \alpha_2^*$

When $\alpha_1 \geq 0$, the same type of argument as above can be applied. Instead of (4.24), the original characteristic equation (4.12), or equivalently,

$$Nz^{N-1}(z-1)^3 + [\alpha_1 z + \alpha_2(z-1)](z^N - 1) = 0 \tag{4.36}$$

should be used. The first step is to note that for $\alpha_1 = 0, 0 < \alpha_2 < \alpha_2^*$, there is a root at $z = 1$ and N roots within unit circle (guaranteed when $0 < \alpha_2 < \alpha_2^*$ by

the previous proof). The root at 1 cancels with that of the numerator, and so the system is stable.

For infinitesimally small α_1 and $0 < \alpha_2 < \alpha_2^*$, we assume that the root z which originally was at 1 now becomes $1 + \epsilon$. Substitute this into (4.36), and ignore power of ϵ higher than the second, we have

$$N\epsilon^2 + [\alpha_1 + (\alpha_1 + \alpha_2)\epsilon][N + \frac{N(N-1)}{2}\epsilon + \begin{pmatrix} N \\ 3 \end{pmatrix}\epsilon^2] = 0 \qquad (4.37)$$

Neglect also the second order terms of ϵ, we finally have

$$\epsilon = \frac{-2\alpha_1}{\alpha_1(N+1) + \alpha_2} < 0 \qquad (4.38)$$

Therefore, with a given $\alpha_2 < \alpha_2^*$, when α_1 is infinitesimally small, the root originally at 1 will move inside and the system will remain stable. When α_1 increases to some value, one of those roots within the unit circle crosses the unit circle. In order to determine that particular value of α_1 for given $\alpha_2 < \alpha_2^*$ (i.e., the boundary of stable region of the system), we apply the transformation $z = e^{j\theta}$ in (4.36). After some manipulation similar to that we have done for the restricted case, we have

$$Ne^{j\frac{N}{2}\theta}(2j\sin\frac{\theta}{2})^3 + (\alpha_1 e^{j\frac{\theta}{2}} + 2j\alpha_2\sin\frac{\theta}{2})(2j\sin\frac{N}{2}\theta) = 0 \qquad (4.39)$$

Note that both the real and imaginary parts of LHS equal to zero, we have

$$\begin{aligned} -8N\cos(N\theta/2)\sin^3(\theta/2) + 2\alpha_1\cos(\theta/2)\sin(N\theta/2) &= 0 \\ 4N\sin^2(\theta/2) - \alpha_1 - 2\alpha_2 &= 0 \end{aligned} \qquad (4.40)$$

Therefore, the boundary of stable region can be found by solving the above two equations numerically. Figure 4.8 shows the stable regions for $N = 4, 6, 8$

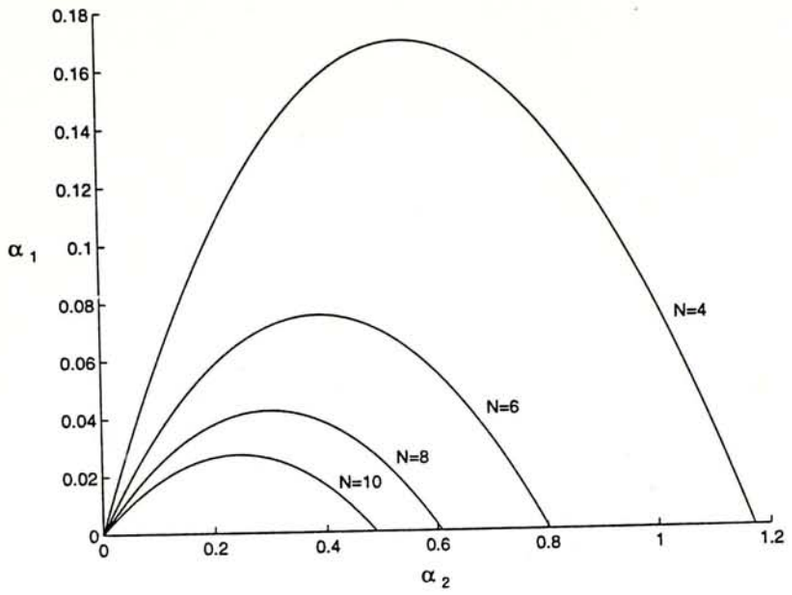


Figure 4.8: Stable regions for (α_1, α_2) for $N = 4, 6, 8$ and 10.

and 10. For each N, the stable region is under the curve. In designing the adaptation controller, we should stay within the stable region when trying out different values of α_1 and α_2. As can be seen, as N increases, the stable region becomes smaller, and there is less freedom in the choice of α_1 and α_2.

4.4.2 Robustness against Coding-mode Switching

Due to coding-mode switching, traffic of a VBR-coded video fluctuates periodically, with period equals to that of the coding-mode switching N. Thus, in the frequency-domain, the video traffic would have components at $0, 1/N, 2/N, 3/N, \ldots$ (in the unit of per-frame-period). This can be verified if we look into Fig. 4.9, [2] which depicts the Discrete Fourier Transform of the MPEG-coded video traffic shown in Fig. 2.3. As can be seen, there are spikes at $1/N$, $2/N$, $3/N$, and so on. These spikes are due to coding-mode switching, while other frequencies can

[2]The frequency has been normalized to the per-frame unit, and correspondingly the maximum frequency is 0.5. There is a large zero-frequency component that is not shown.

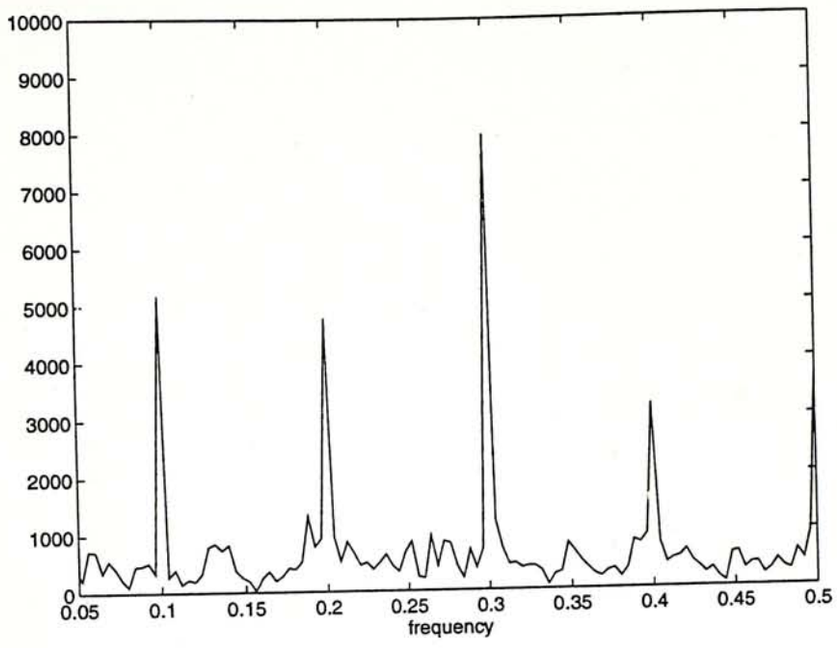


Figure 4.9: Discrete Fourier Transform of the traffic of the sequence $JP2$.

be attributed to the scene content variations.

For our system, we can derive from (4.11) that the transfer function $\Delta r(z)/A(z)$ has zeros at the frequency 0, $1/N$, $2/N$, and so on. This also can be shown by plotting $\Delta r(z)/A(z)$ in Fig. 4.10. Therefore, the $\Delta r[n]$ will show no response to the periodic bit rate fluctuations due to coding-mode switching. In other words, our system can avoid unnecessary image quality fluctuations due to coding-mode switching.

4.4.3 Unit-Step Responses and Unit-Sample Responses

This section studies the effect of a DC change in the video traffic by the unit-step responses (i.e., the system responses under an unit-step input), and the effect of a P/B-frame scene change by the unit-sample responses (i.e., the system responses under an unit-sample input). Particularly, we focus on the quantities $B_0[n]$, $r[n]$

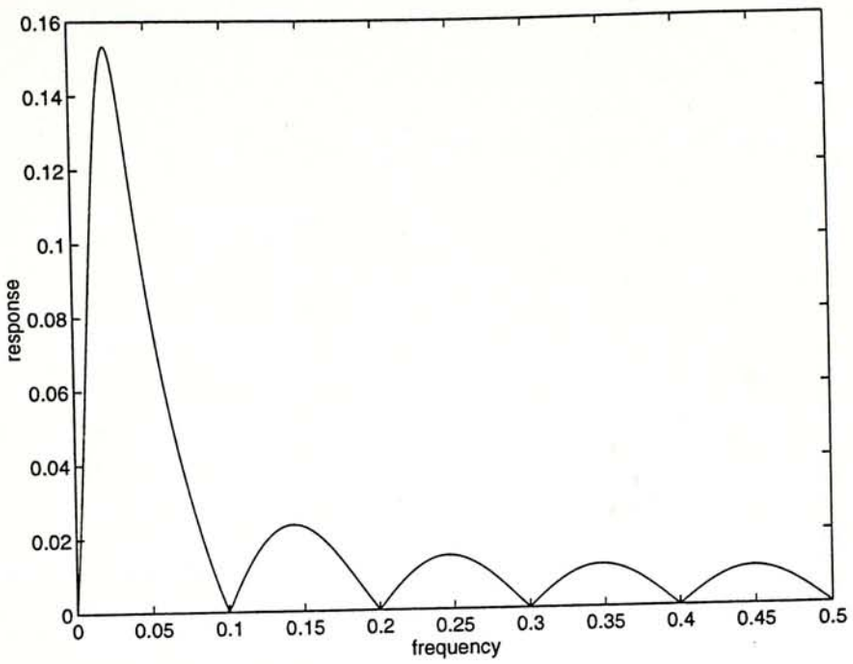


Figure 4.10: Transfer function $\Delta r(z)/A(z)$.

and $\Delta r[n]$,[3] which correspond to the buffer deviation, image-quality degradation, and the change of image quality among successive frames, respectively (see Section 4.3.1). Before that, we shall first further illustrate the significance of the DC compensator (i.e., α_1) in (4.10).

1. Necessity of DC Compensator

In Section 4.3.2.2, we have mentioned that if the feedback compensator does not have the DC compensator (i.e., $\alpha_1 = 0$), then the buffer occupancy cannot be brought to the desired value B whenever there is a DC change in $A[n]$. This can be explained by examining the the transfer function of $B_0[n]$ in (4.11). We note that without the DC compensator (i.e., $\alpha_1 = 0$), the two roots of the denominator at $z = 1$ cancel with those of the numerator, and therefore the

[3]In a linear time-invariant system, $\Delta r[n]$ can by obtained from $r[n]$ by discrete-time "differentiation". Moreover, for unit-step and sample responses, $r[n]$ can be obtained from $B_0[n]$.

response is not zero for DC input. This means that $B_0[n]$ will not decay to zero given a step input. When $\alpha_1 > 0$, on the other hand, there is only one pole at $z = 1$, and the DC response is indeed zero, and therefore $B_0[n]$ will eventually decays to zero given a step input.

2. Unit-Step Responses

We may want to fine-tune the system to fit a individual video sequence, or to operate the system with different performance. Figures 4.11 and 4.12 plot the step responses of the system when $(\alpha_1, \alpha_2) = (0.003, 0.10)$ and $(0.09, 0.17)$ for $N = 10$, respectively. Comparatively, when $(\alpha_1, \alpha_2) = (0.003, 0.10)$, $B_0[n]$ rises to a higher peak and takes a longer time to decay to zero after that; $r[n]$ and $\Delta r[n]$, on other hands, rise to a lower peak. Also, there is no oscillation in all the decay pattern in the $(0.003, 0.10)$ case, but there are in the $(0.009, 0.17)$ case. In other words, in the $(0.003, 0.10)$ case, buffer deviation rises to a higher peak, and hence a larger buffer is needed; in contrast, the worst case image degradation is smaller, and the image-quality transition along the frames is smoother. The former case is a typical weak-feedback setting and the latter a strong one.

Note that the final value of $r[n] = 1$ (and the implied change in image-quality) is a necessity, regardless of the control parameters and strategies, because only then will the buffer input rate equals the output rate, hence avoiding overflow and underflow. Also, note that the $r[n]$ overshoot in the transient behavior is inevitable if it is required that $B_0[n]$ eventually settles to zero: with a step input, initially $B_0[n]$ rises and there is excess bandwidth usage (buffer input exceeds output); and to compensate for this initial overuse of bandwidth, some future frames must be coded at below the final rate on which the system will settle,

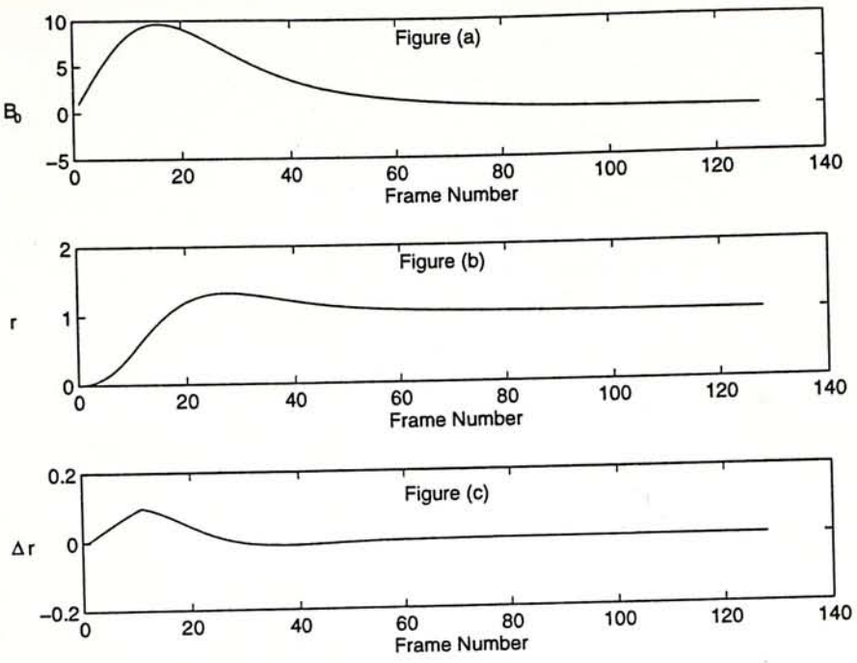


Figure 4.11: Unit-step responses of the system with $(\alpha_1, \alpha_2) = (0.003, 0.10)$: (a) is the buffer deviation $B_0[n]$, (b) is the bit adjustment $r[n]$, and (c) is the change of bit adjustment $\Delta r[n]$.

and $r[n]$ during these frames will overshoot.

To study the effect of adjusting α_1 and α_2 on the unit-step responses, we plot three important parameters as functions of α_1 and α_2: the peak of the buffer deviation $B_0[n]$, B_0^*, is plotted in Fig. 4.13, the peak overshoot of $r[n]$ (i.e., peak of $r[n]-1$), r^*-1, is in Fig. 4.14, and the peak of $\Delta r[n]$, Δr^*, is in Fig. 4.15. [4]

As can be seen, B_0^* decreases when α_1 or α_2 is increased. Therefore, we can reduce the buffer size (and hence delay) by increasing α_1 and α_2. However, when we do so, both r^* and Δr^* increase. This means that the worst case image-quality degradation will be more serious, and the image-quality along the frames will fluctuate more vigorously. In general, there is a trade-off between the image-quality and buffer-occupancy fluctuations. Reference [15] has formulated

[4]Note that in these figures, we only focus on the stable operation region of α_1 and α_2, while the values of the quantities outside the stable region are supposed to be zeros.

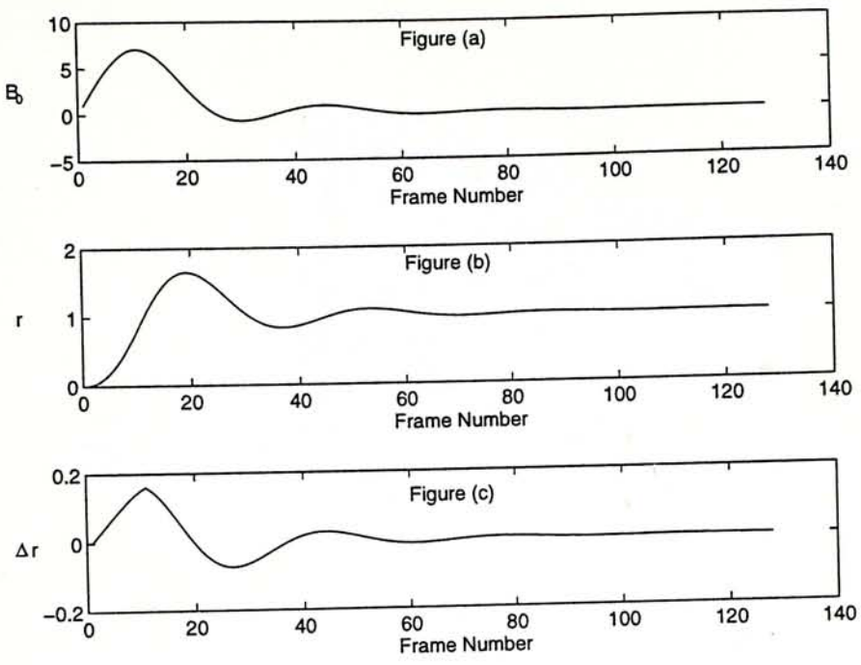


Figure 4.12: Unit-step responses of the system with $(\alpha_1, \alpha_2) = (0.009, 0.17)$: (a) is the buffer deviation $B_0[n]$, (b) is the bit adjustment $r[n]$, and (c) is the change of bit adjustment $\Delta r[n]$.

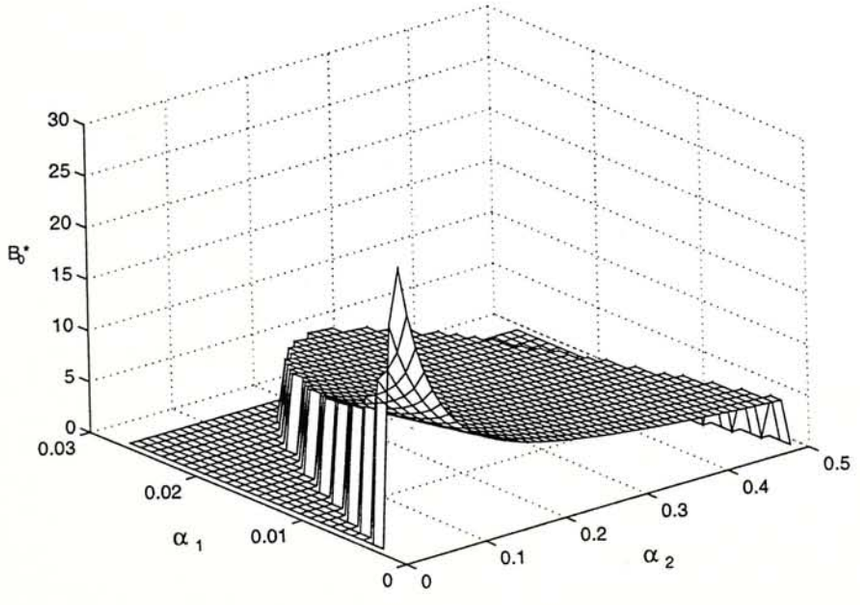


Figure 4.13: Peak of buffer deviation, B_0^*, for unit-step input as a function of α_1 and α_2.

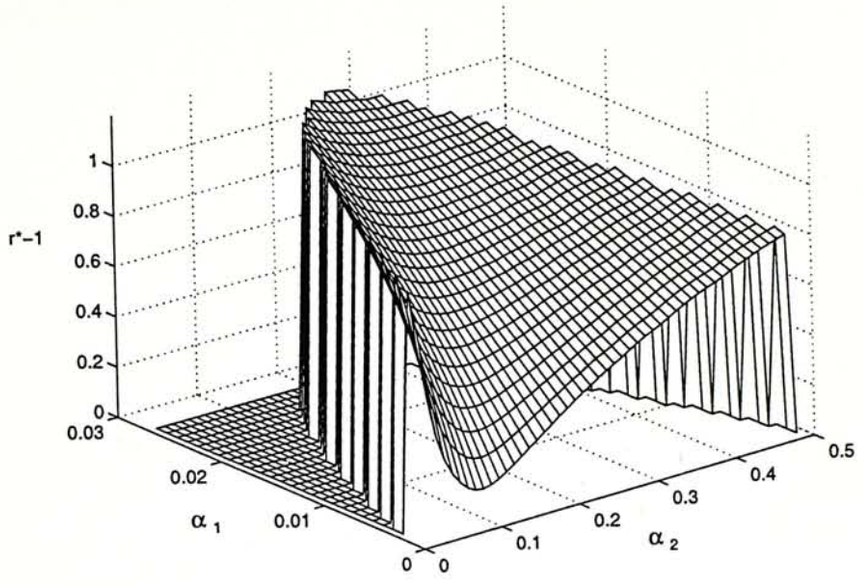


Figure 4.14: Peak overshoot of bit adjustment, $r^* - 1$, for unit-step input as a function of α_1 and α_2.

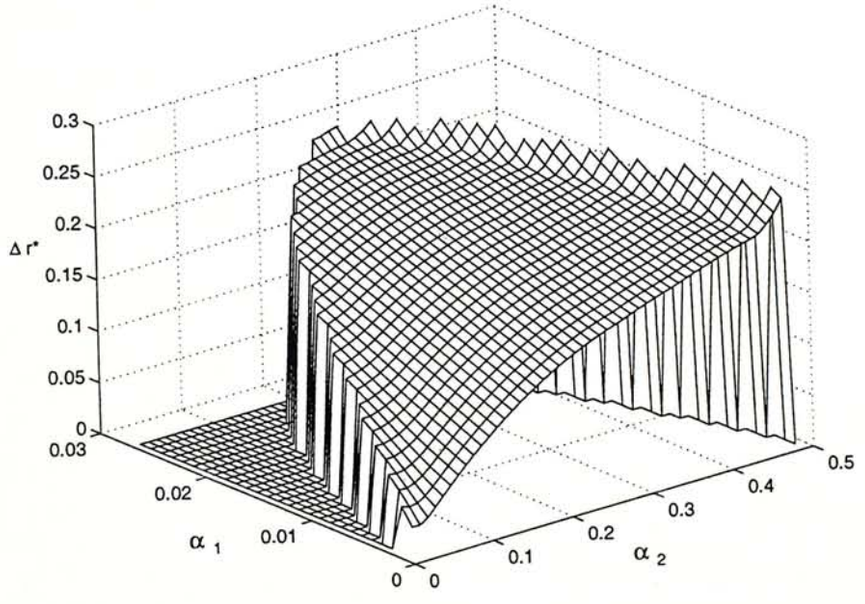


Figure 4.15: Peak of change of bit adjustment, Δr^*, for unit-step input as a function of α_1 and α_2.

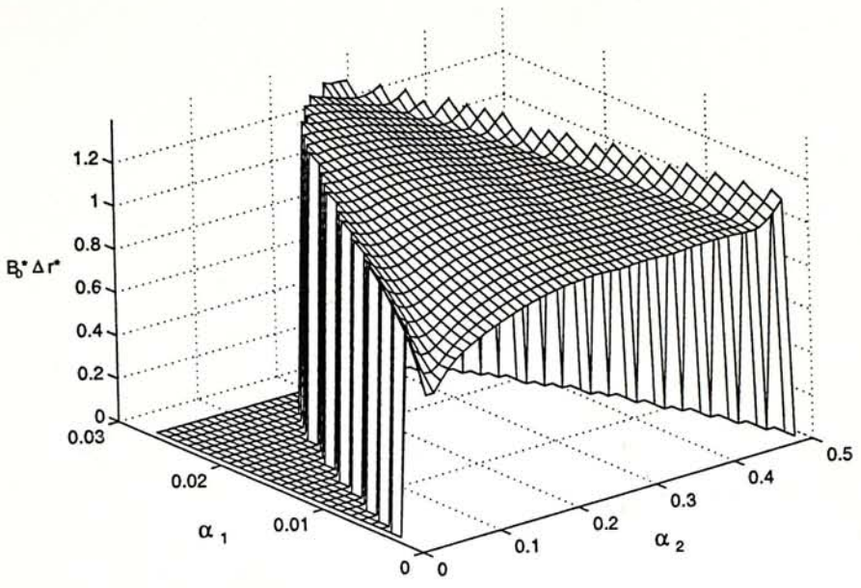


Figure 4.16: Product of B_0^* and Δr^* for unit-step input as a function of α_1 and α_2.

this trade-off as

$$B_0^* \Delta r^* \approx 1$$

This can be justified by plotting the product of B_0^* and Δr^* with varying α_1 and α_2 in Fig. 4.16.

3. Unit-Sample Responses

The unit-sample (impulse) responses of the system with $(\alpha_1, \alpha_2) = (0.003, 0.10)$ and $N = 1$ are shown in Fig. 4.17. In fact, these sample responses can be obtained from the corresponding step responses by discrete-time differentiation.

Since the input is only a unit-sample at $n = 1$, while it is zero for all later frame periods, the peak buffer deviation equals to one (this occurs at $n = 1$) no matter how α_1 and α_2 are set. Moreover, for all cases, the final values of $r[n]$ and $\Delta r[n]$ are zeros. However, different setting of α_1 and α_2 results in different

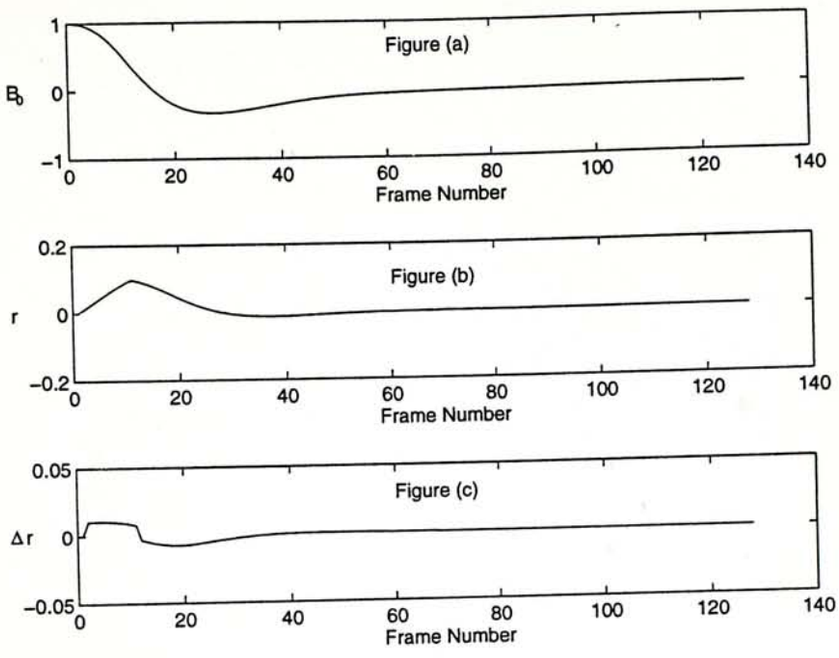


Figure 4.17: Unit-sample responses of the system with $(\alpha_1, \alpha_2) = (0.003, 0.10)$: (a) is the buffer deviation $B_0[n]$, (b) is the bit adjustment $r[n]$, and (c) is the change of bit adjustment $\Delta r[n]$.

peaks of $r[n]$ and $\Delta r[n]$ (i.e., r^* and Δr^*, respectively). Figures 4.18 and 4.19 plot the r^* and Δr^* as functions of α_1 and α_2, respectively. Note that when α_1 or (and) α_2 increase(s), both r^* and Δr^* increase. This means that for sample input, in order to minimize the worst case image degradation and ensure smooth image-quality transition along the frames, small α_1 and α_2 are preferred.

4.5 Implementation

This section describes the implementation of our video traffic adaptation scheme on MPEG-coded video. Specifically, it details how the encoder adapts its output bit rate in response to the control message $\Delta r[n]$.

We assume that the video is first compressed by a standard MPEG encoder

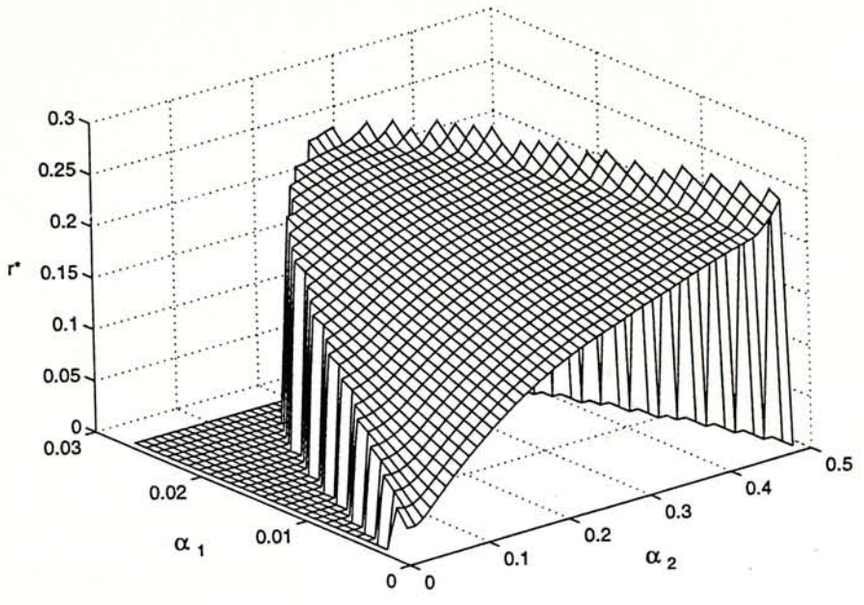


Figure 4.18: Peak of bit adjustment, r^*, for unit-sample input as function of α_1 and α_2.

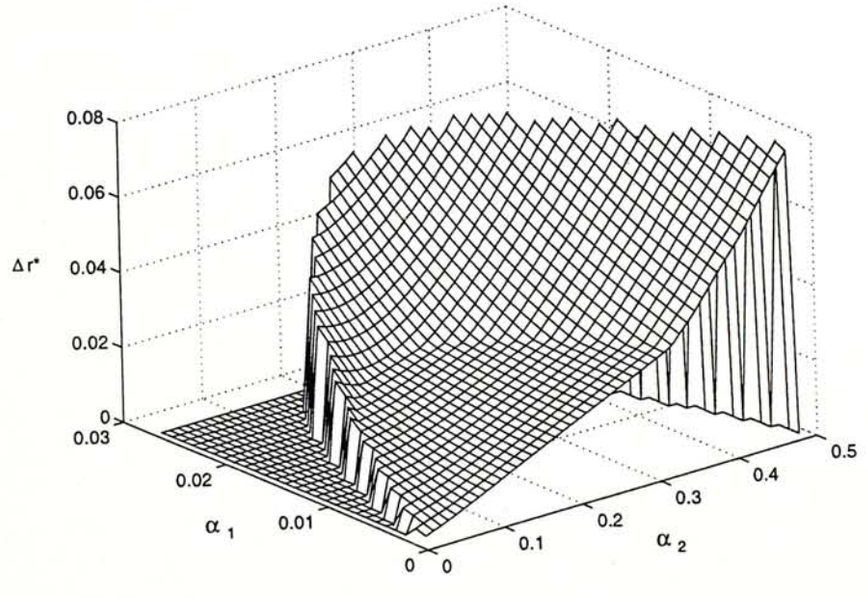


Figure 4.19: Peak of change of bit adjustment, Δr^* for unit-sample input as function of α_1 and α_2.

at pretty good image quality. The video traffic adaptation is then introduced as an add-on process, performed by a secondary encoder: the secondary encoder further compress the video in response to the to the control message from the adaptation controller. The advantage of this system design is that stored video sequences, which are previously compressed by standard MPEG encoders, do not have to be decompressed and then re-compressed during the adaptation process.

In reponse to the control message, the secondary encoder may further reduce the coded bit count of an incoming frame by dropping the AC codewords from the constituent blocks selectively, such that 1) for each block, the codewords dropped are from higher frequencies, and 2) all the blocks finally have almost the same image-quality (according to some metric). However, the header information and the DC codewords will always be transmitted.

To achieve the above goals, we define the "profile" of a frame n, $f_n(D)$, as the number of bits needed for the transmission of the AC codewords as to achieve a distortion of D. For MPEG-coded video, the profile of a frame n can be obtained from the profiles of all the blocks m contained, $B_{mn}(D)$: considering that all the blocks will have the same distortion after adaptation, we have

$$f_n(D) = \sum_m B_{mn}(D) \tag{4.41}$$

where the summation is taken over all blocks in the frame. For $B_{mn}(D)$, the distortion of a block (expressed in dB) is defined as

$$\text{distortion of a block} = 10 \log_{10} \sum_l (s_l - s'_l)^2 \tag{4.42}$$

where s_l is the original value of pixel l within the block, s'_l is the pixel value

after aggregation, and the summation is taken over all pixels l in the block. [5]

The distortion of a block in the frequency domain can be found easily: it is the sum of the amplitude squares of the non-zero components in the discarded AC codewords. Because a codeword is either retained or dropped in its entirety, there is only a finite number of possible distortion levels for each block. In other words, $B_{mn}(D)$ of block m is a staircase-type function and it can be interpreted as the minimum number of bits needed such that the distortion of that block is less than or equal to D. When we add up $B_{mn}(D)$ to obtain $f_n(D)$ as in (4.41), however, many D values are possible. At the frame level, distortion D means that all the constituent blocks have distortion no more than D.

The profiles for an I, a P and a B frames (frames 151, 152 and 153, respectively) in the sequence $JP2$ (see Section 3.4) are plotted in Fig. 4.20. As can be seen, for D between 35 and 45 dB (typical D value resulting from the adaptation experiments that we performed), the profiles for the three types of frames are of similar shape, and hence the assumption in (4.3) is roughly justified. Note that the dB value of distortion is not taken with respect to the signal and there is no significance to its large absolute value: only the relative values of the distortion are important.

After obtaining the profile of a frame n, the operating distortion level $D[n]$ is given by

$$\begin{aligned} f_n(D[n]) &= R'_o[n] - \Delta r[n] \\ &= f_n(D[n-1]) - \Delta r[n] \end{aligned} \tag{4.43}$$

Afterwards, the AC codewords are dropped from each of the blocks according

[5]The reason why we use distortion (i.e., noise energy), instead of SNR, as the metic for image-quality during the adaptation process was described in Section 3.2.2

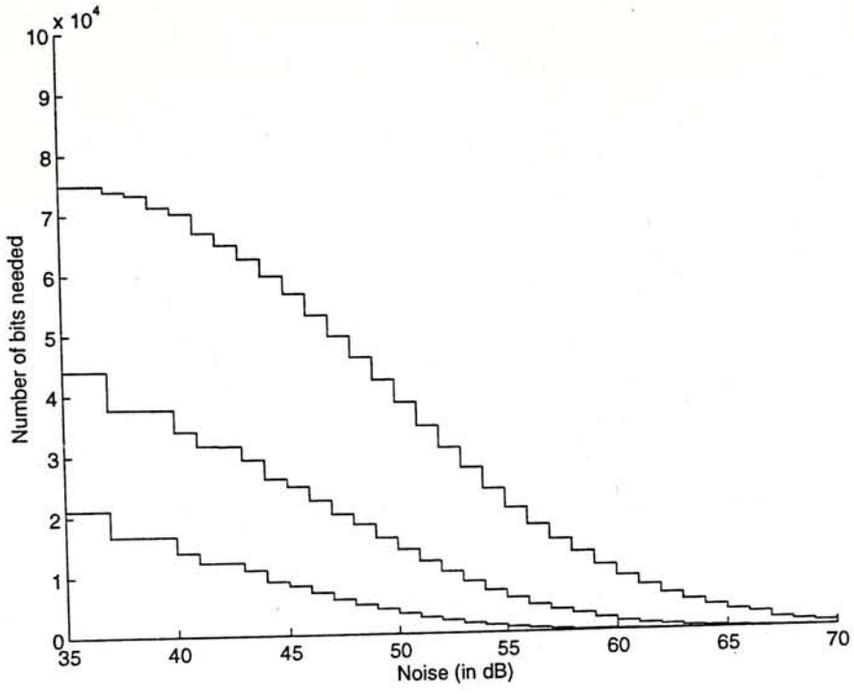


Figure 4.20: The profiles of an I, a P, and a B frames from the sequence $JP2$

to the respective $B_{mn}(D[n])$.

Note that with this implementation, the secondary encoder (for traffic adaptation) can only reduce the bit rate of the MPEG video sequence when the buffer occupancy is high; however, when the buffer occupancy is low, it cannot increase the bit rate of the MPEG video traffic by adding extra data. In other words, at the secondary encoder, bit adjustment $r[n]$ can be positive only, while any negative $r[n]$ is suppressed to zero.

4.6 Experimental Results

This section presents experimental results of transmitting an MPEG video with our traffic adaptation scheme. Specially, we study the overall performance of our adaptation scheme in Section 4.6.1. Section 4.6.2 presents experimental results

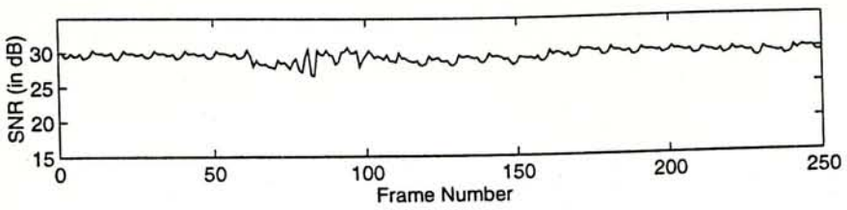


Figure 4.21: The SNR of the sequence $JP2$ before transmission.

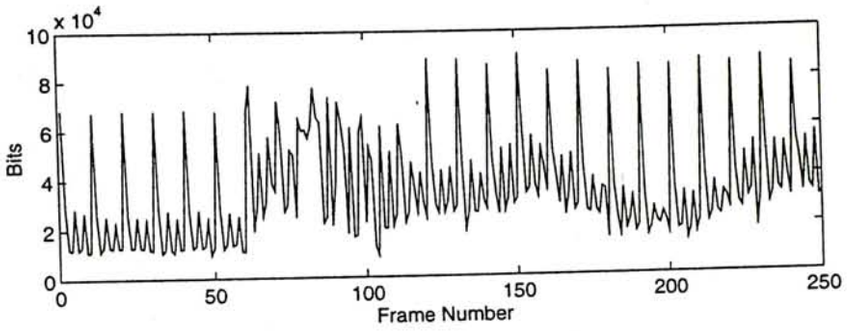


Figure 4.22: The bit rate (in terms of bits per frame) of the sequence $JP2$ before adaptation.

which compare a weak-control setting with a strong-control setting. Section4.6.3 investigates the performance of our adaptation scheme with varying amount of the reserved bandwidth.

The video sequence $JP2$ used in the experiments is 8 seconds in duration. The resolution and frame rate are 320×240 and 30 frames per second, respectively (i.e., quarter size of the NTSC standard). The video sequence was previously coded at good quality (see Fig. 4.21) by a standard MPEG encoder with $N = 10$ and $M = 3$ (see Section 2.1). The video traffic before adaptation in terms of bits per frame was shown in Fig. 4.22, with mean and standard deviation equal to 35.3 and 20.3 kbits, respectively.

4.6.1 Overall Performance of the Adaptation Scheme

The video sequence is then transmitted by a CBR communication channel in three scenarios: 1) with temporal smoothing provided by a buffer, but with no feedback control (i.e., the open-loop approach in Fig. 1.2), 2) with no temporal smoothing at all: at the secondary encoder, we further compress the video sequence to CBR before transmission; and 3) with our traffic adaptation scheme used. In all these scenarios, the output bit rate of the buffer, C (i.e., reserved bandwidth of the CBR channel),set is set to the mean bit rate of the video traffic (i.e., 35.3 kbits per 33.3 ms, or 1.06 Mbits/s); and for scenarios (1) and (3), the size of the buffer, B_{max}, is 333 ms (i.e., 10 frame periods).

The first two approaches do not work well according to our experimental results. For scenario (1), buffer overflow occurs shortly after the DC of the video traffic increases from frame 60 (see Fig. 4.23). Note that this no-feedback scenario corresponds to the case $\alpha_1 = \alpha_2 = 0$ in our adaptation scheme, and hence the system is unstable (see Section 4.4.1). Therefore, unless a bandwidth corresponding to the peak rate of the video traffic is reserved for the CBR channel, buffer overflow may occur when the incoming traffic rate is higher than C for a long period of time. For scenario (2) which does not employ temporal smoothing, the video sequence suffers from serious image-quality degradation (about 5 dB, see Fig. 4.24). This is because for I frames, which intrinsically demand for high bit rates, many frequency components are dropped. Even the P and B frames usually can have all their data transmitted, they also suffer from error propagation (see Section 2.1).

In scenario (3), the video sequence is transmitted with our traffic adaptation scheme. We set $\alpha_1 = 0.003$, $\alpha_2 = 0.10$, and $N = 10$, while the desired buffer

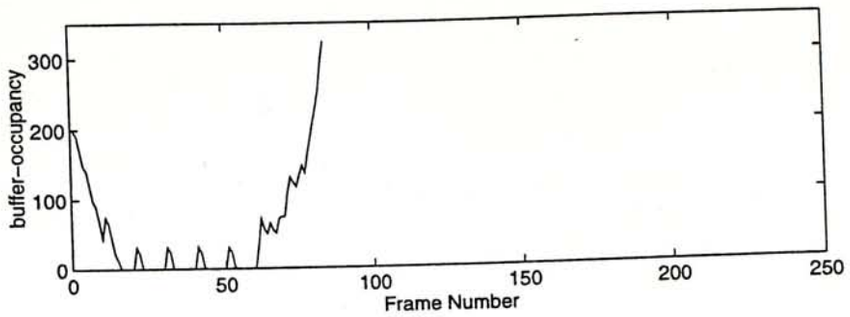


Figure 4.23: The buffer occupancy (in ms) when the sequence $JP2$ is transmitted with temporal smoothing but no feedback control.

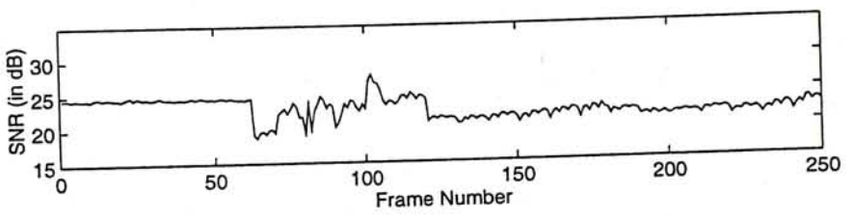


Figure 4.24: The SNR of the sequence $JP2$ after transmitted with no temporal smoothing.

status B equals to $B_{max}/2$ (i.e., 166.5 ms). The traffic of the video sequence after smoothed by our adaptation scheme is shown in Fig. 4.25. The control $\Delta r[n]$, the buffer occupancy deviation $B_0[n]$, and the SNR of the sequence after adaptation are shown in Fig. 4.26 (a), (b) and (c), respectively.

Note that the fluctuations of bit rate in the input traffic (i.e., the video traffic before adaptation) due to coding-mode switching do not cause our adaptation system to perform any undue regulation on the input traffic. However, when the DC of the input traffic is higher than C (e.g., frames 60 to 100, see Fig. 4.22), the secondary encoder drops data selectively (note that the area under $\Delta r[n]$, i.e., the bit reduction $r[n]$, is positive) as to restore the buffer-deviation $B_0[n]$. This results image-quality degradation of the output sequence. However, as the control $\Delta r[n]$ is small, the image-quality transition along the frames is smooth. Comparing with the CBR-compressed sequence (see Fig. 4.24), the sequence

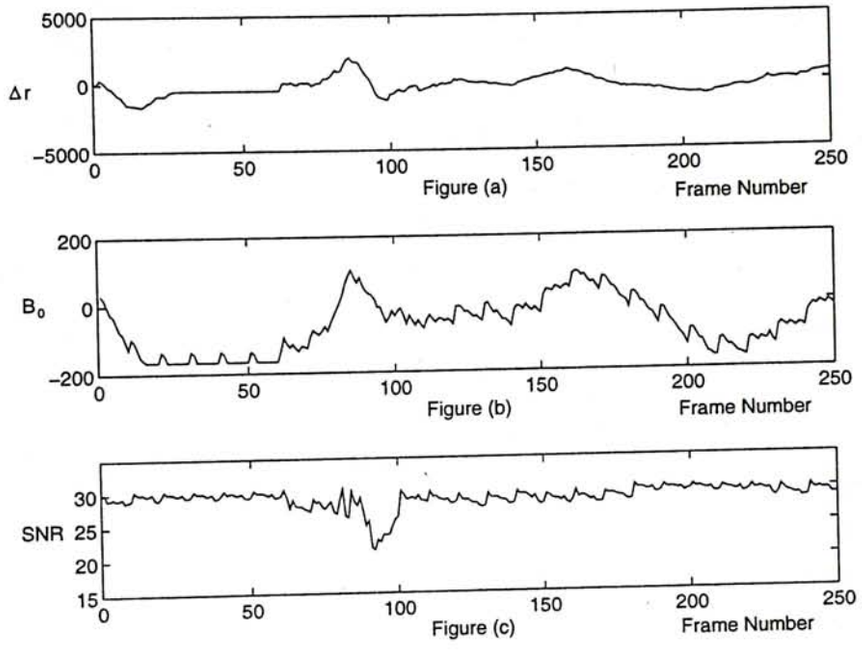


Figure 4.26: Performance of the adaptation scheme with $\alpha_1 = 0.003, \alpha_2 = 0.1$, and C = mean rate of the sequence: (a) the control $\Delta r[n]$ (in bits), (b) the buffer occupancy deviation $B_0[n]$ (in ms), and (c) the SNR (in dB) of the output sequence.

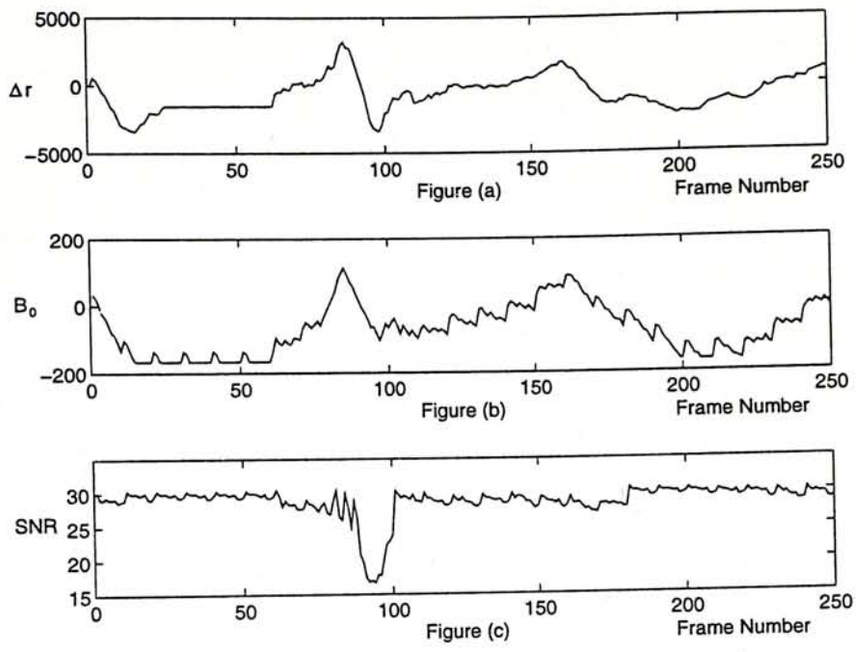


Figure 4.27: Performance of the adaptation scheme with $\alpha_1 = 0.009, \alpha_2 = 0.17$ and C = mean rate of the sequence: (a) the control $\Delta r[n]$ (in bits), (b) the buffer occupancy deviation $B_0[n]$ (in ms), and (c) the SNR (in dB) of the output sequence.

$(\alpha_1, \alpha_2) = (0.009, 0.17)$. The results are depicted in Fig. 4.27. Compared with the results for the (0.003, 0.10) case (especially, from frames 60 to 100), the control $\Delta r[n]$ here is larger, and corresponds to less steady image quality. However, in this strong-control case, the buffer-occupancy deviation is restored more quickly.

4.6.3 Varying Amount of Reserved Bandwidth

In some situation, the mean bit rate of the video sequence is not know before the transmission takes place, especially when the video is generated by a live capture (e.g., for video conference). Therefore, it is also important to verify that our adaptation scheme can perform well even when C is lower than the mean

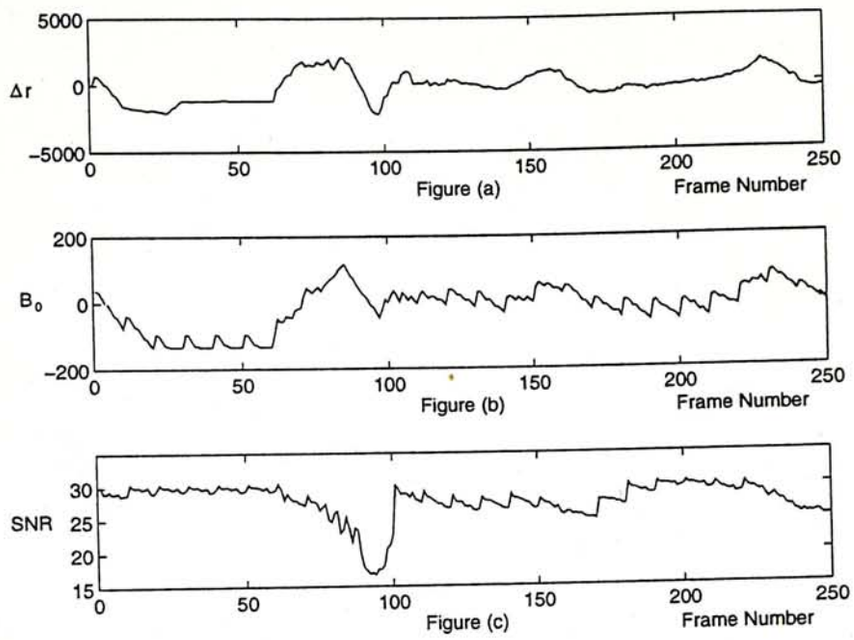


Figure 4.28: Performance of the adaptation scheme with $\alpha_1 = 0.009, \alpha_2 = 0.17$ and $C = 0.8$ of the mean rate: (a) the control $\Delta r[n]$ (in bits), (b) the buffer occupancy deviation $B_0[n]$ (in ms), and (c) the SNR (in dB) of the output sequence.

bit rate of the video sequence.

We transmit the video sequence using our adaptation scheme with $B_{max} =$ 333 ms, $B = B_{max}/2$, $(\alpha_1, \alpha_2) = (0.009, 0.17)$, and C equals to 0.8 of the mean bit rate of the sequence only. The experimental results are depicted in Fig. 4.28. As can be seen, although the overall image quality of the video is worse than that when C equals to the mean rate (see Fig. 4.27), both the image quality and buffer occupancy are still steady.[6]

[6]In fact, the adaptation scheme with such setting can transmit the video sequence with C as low as 0.7 of the mean bit rate only. However, in that case, the image quality of the video is no longer steady.

4.7 Conclusion

This chapter has studied an adaptation scheme for VBR-compressed video traffic which employs temporal smoothing of the video traffic. Specifically, we established a framework for video traffic adaptation based on a linear-feedback control model. Important issues of this adaptation scheme, such as stability, robustness against scene change and coding-mode switching, and the trade-off between image-quality and buffer-occupancy fluctuations, are studied with a control-theoretic approach. We also validated our scheme with experimental results. Compared with previous video traffic adaptation schemes, our framework allows systematic analysis and designs of the adaptation controller, as well as, enables studying the tradeoffs between important characteristics in a systematic and quantifiable manner. More importantly, this control-theoretic framework may opens up many new possibilities for further research.

4.8 Appendix I: Further Research

On the basis of the control-theoretic study of video traffic adaptation we have done in this chapter, this appendix describes some further research possibilities.

Regarding the encoder model in Section 4.3.1.2, we assumed that $\Delta R[n]$ is independent of $\Delta r[i], i \leq n$. In other words, the input and the feedback control of our encoder (see Fig. 4.3) are decoupled. However, in reality, this assumption may be not justified. For instance, when the bits-distortion functions $f_n(D)$ of the frames have different shapes, $\Delta R[n]$ will depend on $D[n-1]$, which in turn depends on $\Delta r[i], i \leq n-1$. Therefore, adaptation (see Fig. 4.1) when the input of the encoder and the feedback control are coupled could be an issue for further

research.

In Section 4.3.1.1, we have mentioned that the adaptation controller can minimize the image-quality fluctuations of the video sequence by minimizing the control $\Delta r[n]$. This is true because for an arbitrary frame n with a given $D[n-1]$, small $\Delta r[n]$ always means that the corresponding change in distortion level $\Delta D[n] = D[n] - D[n-1]$ is also small. However, unless $f_n(D)$ is linear, a given $\Delta r[n]$ does not correspond to a fixed $\Delta D[n]$, because $\frac{df_n}{dD}$ varies with D. Therefore, if we want to ensure smooth image-quality transition along the frames, and set a bound for ΔD, then we will have to take ΔD into account for the computation of the control Δr. How to incorporate the consideration of ΔD into the adaptation system, and how to analyze that system are also worth further research.

Another possibility for further research concerns the output bit rate of the smoothing buffer, $C(t)$. In this chapter, we assumed that $C(t)$ to be constant, because CBR traffic can be guaranteed by networks easily. However, future broadband networks may also be capable of guaranteeing policed VBR traffic (i.e., VBR traffic subjected to some constraints). Suppose the policing of the traffic is done by a leaky bucket [16], the adaptation controller may make use of the tokens (i.e., adjust $C(t)$) and adjust Δr collaboratively, such that comparing with the CBR output scenario, both the image-quality and buffer-occupancy fluctuations can be reduced.

Moreover, as the adaptation controller model described in Section 4.3.2 is based on linear PD control, we believe that some non-linear control, for example, having α_1 and α_1 in equation (4.9) depend on $B_1[n]$, may improve the performance of the adaptation process.

Regarding the implementation of the secondary encoder, the input MPEG video can also be further compressed in the "bit-plane": re-quantize the DCT coefficients with larger quantization factor Q. For MPEG video, many researchers [3, 1] related the quantization factor $Q[n]$ used for coding a frame n, with the corresponding output bit count by

$$X[n] = S[n] \times Q[n] \tag{4.44}$$

where $X[n]$ is called the "global frame complexity " of the frame, and $S[n]$ is the number of bits generated by coding it. As long as this relationship holds, for each frame n, the numbers of output bits with different $Q[n]$ used can be predicted. Since the image quality of a frame n is proportional to $Q[n]$, with respect to the control $\Delta r[n]$, the secondary encoder should re-quantize the video with $Q[n]$ given by

$$\frac{X[n]}{Q[n]} = \frac{X[n]}{Q[n-1]} - \Delta r \tag{4.45}$$

Comparing the frequency-plane approaches described in Sections 4.5, this bit-plane approach needs not calculate the signal energy for every codeword, however, re-quantization of the frames requires additional quantization and VLC coding. Note that it is difficult to ensure the total output bit count in each frame period (from coding the n frames) to be exactly B_t, because 1) the relationship in (4.44) is only an approximate estimation, and 2) for successive choices of Q, the total output bit counts may deviate much. Therefore, the secondary encoder may fail in adapting the video traffic accurately in response to control from the adaptation controller.

corresponding global frame complexity will be updated by (4.48). Moreover, R will be updated as

$$R = R - S_{I,P,B} \tag{4.49}$$

and then allocated among the remaining frames in that GOP base on the weighting of $X_{I,P,B}$. Thus, if the following frame is an I frame, its target bit count, T_I, will be determined by

$$T_I = \frac{X_I}{X_I + N_P X_P + N_B X_B} \times R \tag{4.50}$$

where $N_{P,B}$ are the number of remaining P and B frames, respectively, in the current GOP. Similarly, if the following frame is P or B frame, then $T_{P,B}$ will be determined by

$$T_{P,B} = \frac{X_{P,B}}{N_P X_P + N_B X_B} \times R \tag{4.51}$$

When the target bit count $T_{I,P,B}$ for the following frame is determined, it will be evenly allocated to the MB's in the frame in spite of their individual scene content. In order to minimize the difference between the coded and target bit counts of the frame, the quantization factor Q is adjusted after coding each MB, in proportion to the accumulated difference between the number of coded and target bits. As Q is adjusted after coding each MB, the difference between coded and target bit counts after coding a frame is expected to be small.

The bit allocation algorithm of this scheme is based on the assumption that the global frame complexities are good measures for the intrinsic bit requirements of the frames. However, if this assumption does not hold, unnecessary image quality fluctuations along the frames may result.

Even assuming that the global frame complexities are good measures for the intrinsic bit requirements of the frames, from the viewpoint of image-quality,

Q, ΔQ, is limited to be ± 1. Moreover, in order to reduce the problem caused by scene change, the global frame complexity corresponding to the frame having scene change will be reset after the scene change.

4.9.4 Chen's Adaptation Scheme

The adaptation scheme for MPEG-like encoders proposed by Chen *et.al.* [2] works in a way similar to MPEG's scheme: the residual bits of the current GOP are dynamically allocated to the remaining frames in the GOP with the consideration of the fact that different types of frames have different intrinsic bit requirements.

The bit allocation of Chen's scheme is based on a more global level, called Sub-GOP (SGOP), which is formed by a P frame with its following B frames (i.e., a SGOP consists of M frames, with the first as P frame followed by $M-1$ B frames). Before coding a SGOP, the target bit count allocated for it is determined by

$$T_{SGOP} = \frac{R_{SGOP} \times N_{SGOP} + H_{SM} \times (R_{GOP} - S_I) - B}{N_{SGOP} + H_{SM} \times L} \tag{4.52}$$

where R_{SGOP} and R_{GOP} denote the desired number of a SGOP and GOP, respectively, with definition similar to (4.47). N_{SGOP} is the number of remaining SGOP's in the current GOP, S_I is the number of coded bits of last I frame, B is the deviation of the current buffer occupancy from its desired value, L is the number of SGOP's in a GOP, and H_{SM} is some non-negative integer.

Roughly speaking, T_{SGOP} is the target bit count for the current SGOP so that the buffer can be restored to its expected occupancy at the end of next H_{SM} GOP's (following the current GOP). The integer H_{SM} controls heuristically

the T_{SGOP} variations in the GOP and the instant when the buffer is expected to be restored. When $H_{SM} = 0$, T_{SGOP} for earlier and later SGOP may be drastically different. This may cause improper and unnecessary fluctuations of image-quality along the SGOP's. Larger H_{SM} results in more steady image-quality transition, but at the price of longer period for restoring the buffer (to its expected occupancy), which may imply a larger buffer is needed. It was suggested to have H_{SM} equals to 1. When the target bit count for the current SGOP is determined, all frames in this SGOP will then be coded with the same Q, while Q is adjusted from that for the preceding SGOP in proportion to the difference between the coded and target bit counts for that SGOP. In order to ensure smooth image-quality transition along the SGOP's, ΔQ is limited to some range.

Since every SGOP has the same structure, successive SGOP's should have almost the same intrinsic bit requirements. Therefore, allocating bits in the SGOP level can account for the drastic variations of intrinsic bit requirements of different types of frames, without bothering with the measure for individual frame's intrinsic bit requirements. In other words, this scheme does not rely on the assumption that gobal frame complexities are good measure for the bit requirements as MPEG's scheme does.

Comparing with MPEG's scheme, coding all frames in a SGOP with them same Q can ensure smooth image-quality within the frames, as well as along the frames in the SGOP. However, after coding a SGOP, the difference between the coded and target bit counts may be large, and results in drastic change of Q and hence image-quality among the SGOP's. Although this can be improved by limiting ΔQ, buffer overflow may be caused due to slow response to change

of buffer occupancy.

Bibliography

[1] L. W. Lee, J. F. Wang, J. Y. Lee and C. C. Chen, "On the Error Distribution and Scene Change for the Bit Rate Control of MPEG", *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 3, August 1993.

[2] C-T Chen and A. Wong, "A Self-Governing Rate Buffer Control Strategy for Pseudoconstant Bit Rate Video Coding," *IEEE Transactions on Image Processing*, Vol 2, No. 1, Jan 1993, pp. 50–59.

[3] L. Wang, "Bit Rate Control for Hybrid DPCM/DCT Video Codec", *IEEE Trans. Circuit and Systems for Video Tech.*, Vol. 4, No. 5, October 1994.

[4] H. Watanabe and S. Singhal, "Bit Allocation and Rate Control Based on Human Visual Sensitivity for Interframe Coders", *Proc. ICASSP 1992.*

[5] Benjamin C. Kuo, *Automatic Control Systems, Sixth Edition*, Prentice-Hall, 1991.

[6] N. Ohta, "Packet Video : Modeling and Signal Processing", Artech House. 1994. pp. 164.

[7] C-Y Tse and S. C. Liew, "Video Aggregation : An Integrated Video Compression and Multiplexing Scheme for Broadband Networks," *Proc. IEEE Infocom '95.*

[8] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Commun. Magazine*, May 1994.

[9] D. M. Cohen and D. P. Heyman, "Performance Modeling of Video Teleconferencing in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 6, December 1993.

[10] D. M. Lucantoni, M. F. Neuts and A. R. Reibman, "Methods for Performance Evaluation of VBR Video Traffic Models", *IEEE/ACM Trans. Networking*, Vol 2, No. 2, April 1994.

[11] N. M. Marafih, Y-Q. Zhang and R. L. Pickholtz, "Modeling and Queuing Analysis of Variable-Bit-Rate Coded Video Sources in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 4, No. 2, April 1994.

[12] B. Melamed, D. Raychaudhuri, B. Sengupta and J. Zdepski, *TES-Based Traffic Modeling for Performance Evaluation of Integrated Networks, Proc. IEEE Infocom 1992.*

[13] S. M. Shinners, *Modern Control System Theory and Design*, John Wiley and Sons, 1992.

[14] W. McC. Siebert, *Circuit, Signals, and Systems*, MIT Press, 1986.

Chapter 5

Conclusion

This thesis investigated VBR-CBR adaptation of video sequences: the output traffic of a VBR-encoder is adapted to CBR, so that it can be delivered over the network using a CBR channel. Note that this VBR-CBR video adaptation can achieve both the advantages of steady image quality of the video and simple network operation simultaneously. In general, there are two possibilities for the VBR-CBR video adaptation: 1) spatial smoothing among a number of video sequences at a particular time moment, and 2) temporal smoothing within a single video sequence at different time moments.

For spatial smoothing, we investigated a concept called video aggregation, which is defined as the integration of compression and statistical multiplexing of video information. It has been shown experimentally (based on the objective SNR measure and subjective observation of image quality) that video aggregation can provide better image quality than the scenario which does the compression and multiplexing processes separately. In particular, two important goals are achieved: 1) smooth and good image quality for the frames of each video

session, and 2) fairness of image quality among the video sessions. Experimental results show that for the same image-quality requirements, video aggregation can reduce the bandwidth usage by 25%.

For temporal smoothing, we established a framework for video traffic adaptation based on a linear-feedback control model. Important issues of this adaptation scheme, such as stability, robustness against scene change and coding-mode switching, and the trade-off between image-quality and buffer-occupancy fluctuations, are studied in a control-theoretic approach. Compared with previous video traffic adaptation schemes, our model can be designed and analyzed in a systemic and quantifiable manner. More importantly, this control-theoretic framework may open up many new possibilities for further research.

Although spatial and temporal smoothing of video traffic are studied separately in the thesis, they are complementary to each other: a number of video streams can be first smoothed to less bursty by spatial smoothing, afterwards, this multiplexed traffic is further smoothed to CBR by temporal smoothing. Integration of spatial and temporal smoothing as a hybrid adaptation system could be an issue for further research.

In the thesis, all the discussions on both spatial and temporal video traffic smoothing are limited to CBR output, however, they can be extended to VBR output, provided that the broadband network is capable of guaranteeing policed VBR traffic. Moreover, most of the discussions are applicable to generic VBR video compression schemes, even though all our implementations for experiments are conformed on the MPEG compression standard.

Bibliography

[1] M. De Prycker, *Asynchronous Transfer Mode : Solution for Broadband ISDN*, Ellis Horwood, 1993.

[2] N. Ohta, "Packet Video : Modeling and Signal Processing", Artech House. 1994. pp. 164.

[3] D. Le Gall, "MPEG : A Video Compression Standard for Multimedia Applications", *Commun. of the ACM*, Vol. 34, pp.47–58, April 1991.

[4] C-Y Tse and S. C. Liew, "Video Aggregation : An Integrated Video Compression and Multiplexing Scheme for Broadband Networks," *Proc. IEEE Infocom '95.*

[5] P. Pancha and M. El Zarki, "MPEG Coding for Variable Bit Rate Video Transmission", *IEEE Commun. Magazine*, May 1994.

[6] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 3, June 1993.

[7] M. Ghanbari, "Two-Layer Coding of Video Signals for VBR Networks", *IEEE J. Selected Areas in Commun.*, Vol. 7, No. 5, June 1989.

[8] F. Kishino, K. Manabe, Y. Hayashi and H. Yasuda, "Variable Bit-Rate Coding of Video Signals for ATM Networks", *IEEE J. Selected Areas in commun.*, Vol. 7, No. 5, June 1989.

[9] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta and J. Hill, "Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks", *Proc. IEEE Infocom 93*, pp. 919–926.

[10] S. S. Dixit and P. Skelly, "Video Traffic Smoothing and ATM Multiplexer Performance", *Proc. IEEE Globecom 91*, pp. 0239–0243.

[11] R. Coellco and S. Tohme, "Video Coding Mechanism to Predict Video Traffic in ATM Network", *IEEE Globecom 93*, pp. 447–450.

[12] D. M. Cohen and D. P. Heyman, "Performance Modeling of Video Teleconferencing in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 3, No. 6, December 1993.

[13] N. M. Marafih, Y-Q. Zhang and R. L. Pickholtz, "Modeling and Queuing Analysis of Variable-Bit-Rate Coded Video Sources in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 4, No. 2, April 1994.

[14] L. W. Lee, J. F. Wang, J. Y. Lee and C. C. Chen, "On the Error Distribution and Scene Change for the Bit Rate Control of MPEG", *IEEE Trans. on Consumer Electronics*, Vol. 39, No. 3, August 1993.

[15] C-T Chen and A. Wong, "A Self-Governing Rate Buffer Control Strategy for Pseudoconstant Bit Rate Video Coding," *IEEE Transactions on Image Processing*, Vol 2, No. 1, Jan 1993, pp. 50–59.

[16] L. Wang, "Bit Rate Control for Hybrid DPCM/DCT Video Codec", *IEEE Trans. Circuit and Systems for Video Tech.*, Vol. 4, No. 5, October 1994.

[17] H. Watanabe and S. Singhal, "Bit Allocation and Rate Control Based on Human Visual Sensitivity for Interframe Coders", *Proc. ICASSP 1992.*

[18] M. Kawashima, C. Chen, F. Jeng and S. Singhal, "Adaptation of the MPEG Video-Coding Algorithm to Network Applications", *IEEE Trans. Circuit and Systems for Video Tech.*, Vol. 3, No. 4, August 1993.

[19] K. R. Rao and P. Yip, *Discrete Cosine Transform : Algorithm, Advantages, and Applications*, Academic Press, Inc., 1990, pp. 170.

[20] W. Verbiest, L. Pinnoo and B. Voeten, "The Impact of the ATM Concept on Video Coding", *IEEE JSAC*, Vol. 6, No. 9, December 1988.

[21] H. Gharavi and M. H. Partovi, "Video Coding and Distribution over ATM for Multipoint Teleconferencing", *Proc. IEEE Globecom 1993.*

[22] D. Deloddere, W. Verbiest and H. Verhille, "Interactive Video On Demand", *IEEE Comm. Magazine*, May 1994.

[23] G. Ramamurthy and B. Sengupta, "Modeling and Analysis of a Variable Bit Rate Video Multiplexer", *Proc. IEEE Infocom 1992.*

[24] M. Ghanbari and C. J. Hughes, "Packing Coded Video Signals into ATM Cells", *IEEE/ACM Trans. on Networking*, Vol. 1, No. 5, October 1993.

[25] Benjamin C. Kuo, *Automatic Control Systems, Sixth Edition*, Prentice-Hall, 1991.

[26] N. M. Marafih, Y-Q. Zhang and R. L. Pickholtz, "Modeling and Queuing Analysis of Variable-Bit-Rate Coded Video Sources in ATM Networks", *IEEE Trans. on Circuit and Systems for Video Tech.*, Vol. 4, No. 2, April 1994.

[27] B. Melamed, D. Raychaudhuri, B. Sengupta and J. Zdepski, *TES-Based Traffic Modeling for Performance Evaluation of Integrated Networks, Proc. IEEE Infocom 1992.*

[28] S. M. Shinners, *Modern Control System Theory and Design*, John Wiley and Sons, 1992.

[29] W. McC. Siebert, *Circuit, Signals, and Systems*, MIT Press, 1986.

[30] S. C. Liew and C. Y. Tse, "A Control-Theoretic Approach to Adapting VBR Compressed Video for Transport over a CBR Communications Channel", *Submitted to IEEE Infocom 1996.*