

A Control-Theoretic Approach to Adapting VBR Compressed Video for Transport Over a CBR Communications Channel

Soung C. Liew, *Senior Member, IEEE*, and Derek Chi-yin Tse, *Member, IEEE*

Abstract—Future broad-band communications networks are expected to be dominated by video and image traffic. Variable bit-rate (VBR) video compression is generally preferred to constant bit-rate (CBR) compression because constant image quality can be provided. In contrast, CBR transport is preferred to VBR transport from the networking standpoint because of its simplicity. This paper studies the important issue of adapting VBR-compressed video for transport over a CBR channel. We focus on temporal traffic smoothing using an elastic buffer. The target image quality and the output rate of the video encoder is controlled by feedback based on the buffer-occupancy level. Previous adaptation schemes are not readily analyzable. An analyzable control-theoretic adaptation framework is proposed. It allows systematic and quantitative investigation of issues such as stability, robustness against scene changes, robustness against image-quality oscillations due to coding-mode switching, and tradeoffs between image-quality and buffer-occupancy (delay) fluctuations. Perhaps more importantly, the framework opens up many new possibilities for further research.

Index Terms—Rate control, traffic adaptation, video-buffer control, video compression, video transport.

I. INTRODUCTION

FUTURE broad-band communications networks are expected to carry information traffic generated by a wide variety of services and applications. Video, however, is likely to dominate because it typically requires much more bandwidth than other media. 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 [1]. VBR compression is generally preferred because constant image quality can be provided.

CBR and VBR transport, as distinct from compression, refers to using CBR and VBR channels, respectively, for the

transport of data. Transport using CBR channels has many advantages from the networking standpoint. Since the data rate is constant, bandwidth allocation and tariff for network usage are simple. It is also more straightforward for the network to multiplex the traffic from several CBR channels onto a common physical link.

Given the desirability of VBR compression and CBR transport, an issue is how to adapt VBR-compressed traffic for CBR transport; in other words, how to adaptively code and smooth the video sequence so that it can be delivered over the network using a CBR channel. There are two possibilities for traffic smoothing: 1) spatial smoothing and 2) temporal smoothing. In spatial smoothing, several high-quality VBR-coded streams are considered together. The idea is that the bandwidth-hungry streams may “borrow” bandwidth from less demanding streams. Although individually the streams are VBR, as a whole the traffic is made to be CBR. In this way, both the advantages of VBR compression and CBR transport can be achieved simultaneously. For instance, [2] proposes a technique called *aggregation* in which compressed data in excess of the CBR channel bandwidth is discarded in a way that the degradation is minimal and fair among all streams during the adaptation process.

This paper concerns adaptation by temporal smoothing. The general issues will be discussed in Section II. Section III presents a generic adaptation controller based on the control-theoretic approach. We show that the control mechanism can be studied and analyzed on a rigorous basis in Section IV. Section V is devoted to a graphical analysis of the tradeoffs between different objectives of the adaptation controller. A specific implementation of the theoretical adaptation controller and the experimental results based on it are given in Section VI. Possible extensions of this work are discussed in the conclusion in Section VII.

II. ADAPTATION BY TEMPORAL SMOOTHING

Many VBR compression schemes use interframe and interpolative coding that exploit the strong correlation among successive frames of a video sequence to reduce the bandwidth requirement. This is the case, for instance, in MPEG¹ coding [2]–[4]. Intraframe-coded frames (called I frames) are coded

¹MPEG stands for Moving Picture Experts Group. MPEG coding has been developed as a standard of video compression and is fast becoming the coding scheme of choice among product developers.

Manuscript received July 18, 1995; revised June 20, 1997; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. El Zarki. This work was supported by the RGC Earmarked Grant of the Hong Kong University and Polytechnic Grant Council under Grant CUHK 336/96E.

S. C. Liew is with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong (e-mail: soung@ie.cuhk.edu.hk).

D. C. Tse was with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, NT, Hong Kong. He is now with Newbridge Networks (Asia), Ltd., Hong Kong.

Publisher Item Identifier S 1063-6692(98)01452-6.

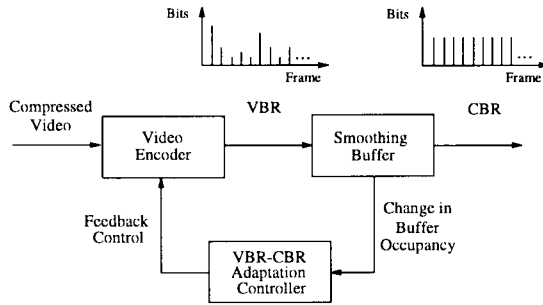


Fig. 1. A general VBR–CBR adaptation model that employs temporal smoothing.

independently without referring to other frames. Interframe-coded frames (called P frames), on the other hand, are coded with reference to a preceding I or P frame so that information that can be derived (or predicted according to an algorithm) from this preceding frame needs not be coded; only information that cannot be derived is coded and sent. Interpolative frames (called B frames) are coded in a similar way; however, both the preceding and the following P or I frames are used as the reference. Upon reception of a P or B frame, the decoder can reconstruct the original information by piecing together the received information and the information that it can derive from the reference frame(s) by using the same algorithm as the encoder. A P or B frame would generally require fewer bits than an I frame. However, to prevent a phenomenon called error propagation and to facilitate random access of the video [4], [5], a frame is usually intraframe-coded periodically. Thus, an example of the coding modes of a video sequence could be

IBBPBBPBBPIBBPBBPBBPI...

in which the period of the coding-mode cycle is $N = 10$. We would therefore expect the number of bits required to change from frame to frame and a peak to occur every N frames.

In temporal smoothing, the 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. Fig. 1 shows a general VBR–CBR adaptation model that employs temporal smoothing. There are three components in the system: the video encoder, the smoothing buffer, and the VBR–CBR adaptation controller. The operation is slotted into time intervals of Δt duration. In each Δt 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 buffer occupancy reflects the difference between the numbers of input and output bits. Based on the buffer occupancy, the adaptation controller controls the future output rate of the encoder via a feedback path.

Different control schemes may differ in terms of the feedback time interval Δt . In this paper, we choose Δt to be a frame period. The majority of the schemes proposed to date assume that the feedback is specified in terms of a change in the quantization factor [5]–[7] ΔQ used in the compression

process. In this paper, we assume that the feedback is specified in terms of a change in the bit rate Δr . This is a more general model: we can map Δr into ΔQ for encoders that adjust the quantization factor, or map Δr into other parameters for encoders that employ other bit-adjustment mechanisms.

For instance, to map Δr into ΔQ in MPEG video, the complexity measure of MPEG TM5 [8] can be used. It states that the complexity X of a video frame equals the product RQ , where R is the number of bits and Q is the average quantization scale. From this, we can obtain $\Delta Q = (X/R^2)\Delta r$

Central to the model in Fig. 1 is the design of the adaptation controller. For its design, we need to have an idea on what sort of input is to be expected. Fig. 2(a) shows the bits per frame of an MPEG-coded video sequence (without being subjected to rate control of adaptation). For the study of adaptation, we may model a typical video-sequence “signal” as being composed of several components. For illustration, we may decompose the signal in Fig. 2(a) into four components. Fig. 2(b) is the DC component that changes only once in a while. A DC change may correspond to a long-term change in scene complexity. A very drastic DC change, for instance, could happen when a movie steps up from a slow-paced plot to an action-filled plot. Fig. 2(c) is the “impulse” component that corresponds to one-time scene changes during P or B frames. This could happen, for example, when the angle of shooting changes during a P or B frame, rendering the reference frames not useful in reducing the bit requirement of the frame. For MPEG-like compression that makes use of different coding modes in a cyclic manner, we also have the coding-mode switching component, which is depicted in Fig. 2(d). Finally, Fig. 2(e) is the component associated with small random variations within the same scene.

The adaptation controller to be described in this paper is a linear controller. This allows us to appeal to the superposition principle for its study. That is, we may study the isolated system response to each of the input signal components independently. The overall response will be the sum of the responses of the individual input components.

Unlike many other studies that attempt to characterize video traffic stochastically [1], [9]–[12] in a detailed manner, we take the standpoint that a universal statistical traffic model is intrinsically difficult. This is particularly true for movies, which tend to have widely varying statistics. For our system to be generally applicable, we attempt to capture only some simple but fundamental characteristics of video traffic.

With respect to the above signal decomposition, let us examine the main objectives of the adaptation controller. Since the output channel of the adaptation system is CBR, when the bandwidth (the DC component) of the video sequence becomes higher than can be sustained over a long period of time, the adaptation controller must inform the encoder to reduce its bit rate by operating at a lower image-quality target. By the same token, when the encoder produces fewer bits than can be transmitted over a long stretch of time, the image quality can be raised so as to make full use of the channel bandwidth.

Meanwhile, short-term fluctuations (the impulse, coding-mode switching, and small random components) in bandwidth should be smoothed out without substantial fluctuations in

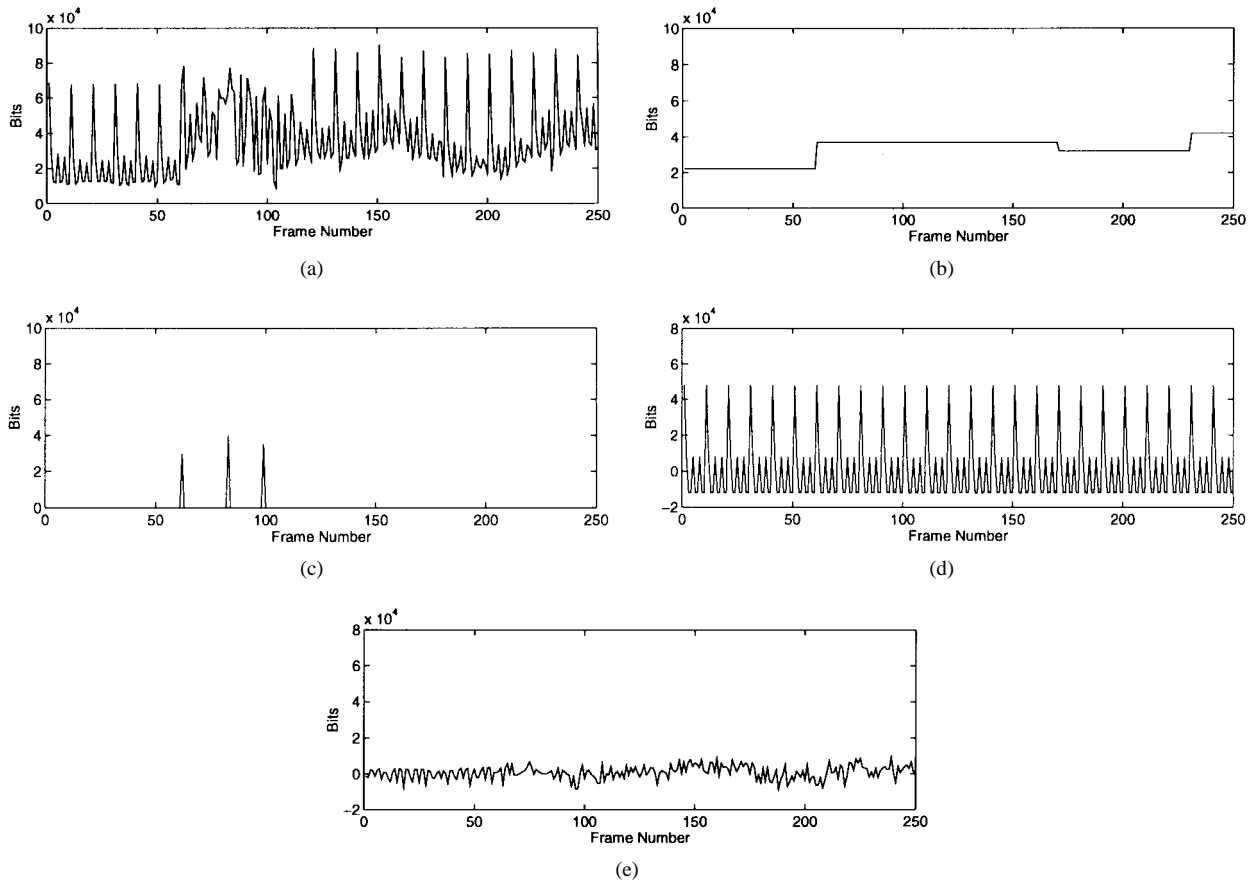


Fig. 2. Decomposition of the traffic of an MPEG-coded sequence. (a) Traffic of an MPEG-coded sequence. (b) The DC component. (c) The scene change component. (d) The code-mode component. (e) The random component.

image quality and buffer occupancy. In short, the adaptation controller must meet three objectives as far as its “transient” behavior is concerned:

- minimize fluctuations in image quality;
- minimize buffer-occupancy fluctuations;
- avoid unnecessary control of the encoder due to switching of coding modes.

The first objective is obvious since frame-to-frame fluctuations of image quality can be visually disturbing. There are two reasons for the desirability of small buffer-occupancy fluctuations. The first is that small fluctuations mean a smaller buffer can be used without buffer overflow or underflow being encountered. Overflow and underflow are undesirable because when the buffer overflows, important data may be discarded and this results in serious image degradation, and when the buffer underflows, the capacity of the CBR channel is not fully utilized and this means that the image could have been encoded with higher quality. The second motivation for small buffer fluctuations is that even if a large buffer were available, larger buffer-occupancy fluctuations result in large end-to-end delay jitter from the sender to the receiver.

The third objective applies to compression schemes that employ coding-mode switching. The buffer occupancy would fluctuate in accordance with the coding mode even if there were no scene changes and the scene complexity were roughly the same for successive frames. These fluctuations are “nat-

ural” and should not cause the adaptation controller to apply undue control on the encoder that might lead to fluctuations in visual quality.

Most approaches proposed to date [5]–[7] are *ad hoc* and lack an analytical framework. For instance, there is generally a tradeoff between the smoothness-of-quality and buffer fluctuations criteria (see Section V), and without an analytical framework, these approaches do not lend themselves to a systematic study of this tradeoff. Although some previous work also deals with the third criterion of robustness against coding-mode switching, lacking is a quantitative investigation showing that the schemes are robust against all coding-mode switching patterns. In short, optimization and design with these approaches is more of an art than science. This paper, in contrast, attempts a systematic approach. It draws on the rich results from control theory. Desired properties and tradeoffs can be studied in a concrete and quantifiable manner.

III. LINEAR FEEDBACK CONTROLLER MODEL

This section describes a generic linear-feedback controller model. Subsection A describes the operation of the encoder and proposes an analytical encoder model. Subsection B describes a filter that avoids unnecessary image-quality adjustments triggered by coding-mode switching. The feedback control and overall system responses are discussed in Section III-C.

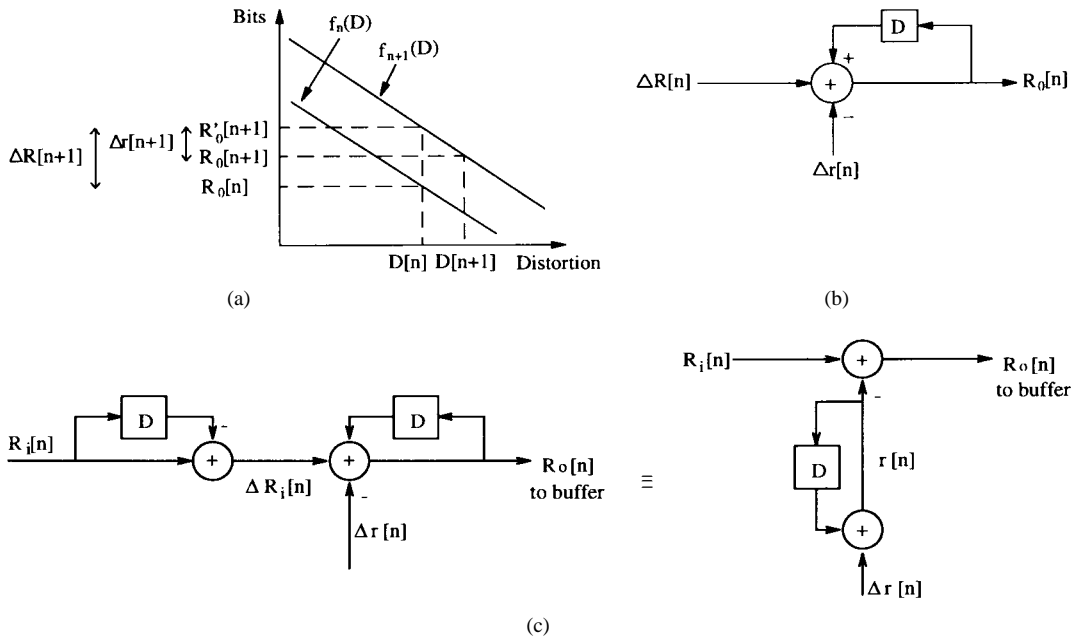


Fig. 3. The encoder operation and encoder analytical model. (a) Bit adjustment at the encoder. (b) The encoder model. (c) The modified encoder model before and after simplification.

A. Encoder Operation and a Linear Encoder Model

We propose an abstract but generic encoder model here. The experiments in Section VI are based on a specific implementation that falls under the abstract framework. For each frame, we assume that the encoder can code it with a certain image-quality objective. In addition, in the vicinity of a given distortion level, the encoder can either estimate or compute explicitly a bits-versus-distortion function $f_n(D)$ that describes the number of bits required to achieve a distortion of D . This function depends on the scene contents as well as the coding mode. The encoder interacts with the adaptation controller as follows. With reference to Fig. 3(a), suppose that for frame n the number of bits output by the encoder is $R_o[n]$ and this corresponds to a distortion of $D[n]$. To maintain the same distortion (hence no image-quality fluctuation) in frame $n+1$, let the number of bits needed in frame $n+1$ be $R'_o[n+1]$. Although the distortion levels are the same, the values of $R'_o[n+1]$ and $R_o[n]$ are, in general, different. We can express $R'_o[n+1]$ as

$$R'_o[n+1] = R_o[n] + \Delta R[n+1] \quad (1)$$

where $\Delta R[n+1]$ is the difference between the numbers of bits needed in frame $n+1$ and frame n [see Fig. 3(a)] for the same distortion.

The encoder will output $R'_o[n+1]$ bits and maintain the same distortion level in the absence of feedback from the adaptation controller. The feedback, which may change the operating distortion level, is specified in terms of $\Delta r[n+1]$, the number of bits to be reduced from $R'_o[n+1]$. The encoder then codes frame $n+1$ with the objective of outputting

$$R_o[n+1] = R'_o[n+1] - \Delta r[n+1] \text{ bits.} \quad (2)$$

For small $\Delta r[n+1]$ the corresponding change in distortion level $\Delta D[n+1] = D[n+1] - D[n]$ is also small. There-

fore, the adaptation controller can minimize the image-quality fluctuation by minimizing $\Delta r[n+1]$.

Fig. 3(b) is a model of the encoder that describes the output traffic. The input is $\Delta R[n]$, the change in number of bits in order to keep the same distortion as the preceding frame's, and the output is $R_o[n]$. With the encoder model, the behavior of an adaptation-controller design can be studied by inputting different forms of $\Delta R[n]$. To simulate a long-term change in scene complexity, we could use a sample (impulse) input. With respect to Fig. 3(a), this corresponds to $f_n(D)$ moving vertically. To simulate a one-time scene change which happens during a P or B frame, we could use an input that consists of a positive (negative) sample immediately followed by a negative (positive) sample. To simulate coding-mode switching, we could use a periodic input with period N . The adaptation system to be described is linear. By appealing to the superposition principle, we can study the effectiveness or robustness of the system against scene changes and coding-mode switching separately.

One assumption that we shall make in our analytical study in Section IV is that $\Delta R[n]$, the input, is independent of the past and present feedback control $\Delta r[i]$, $i \leq n$. Strictly speaking, this assumption will be valid only if the bits-versus-distortion curves of successive frames are offsets of one another in the vertical direction. This approximation is needed so that the analysis is tractable.

It is customary in the study of the transient behavior of linear systems to use step and sample inputs rather than the sample and positive-negative dual samples described above (for simulating scene changes). We note that the latter is simply the discrete-time differentiation of the former. Thus, we can define a new parameter $R_i[n]$

$$R_i[n] - R_i[n-1] = \Delta R[n] \quad (3)$$

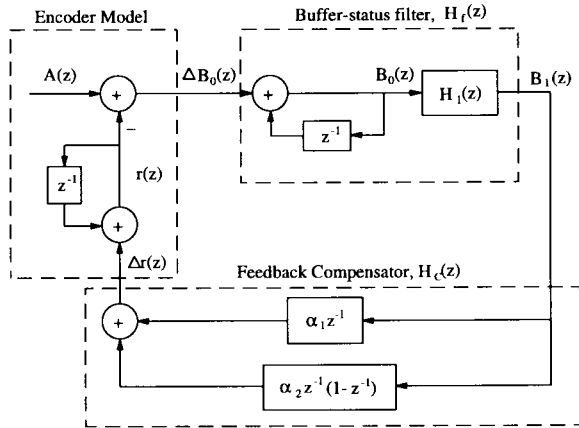


Fig. 4. The z -domain block diagram of the encoder model and the adaptation controller.

and replace $\Delta R[n]$ by $R_i[n]$ followed by a differentiator, as shown in the modified encoded model in Fig. 3(c). Then, step and sample inputs for $R_i[n]$ can be used to simulate scene changes. It should be noted that the model in Fig. 3(c) is for analytical purposes only, and the actual encoder keeps track of neither $R_i[n]$ nor $r[n]$.

B. Buffer-Occupancy Filter

Fig. 4 is a z -domain [13] block diagram that shows the encoder model being connected to the adaptation controller, which consists of two parts: the buffer-occupancy filter and the feedback compensator. In our notation, the z -transform of a time-domain signal $X[n]$ is $X(z) = \sum_{n=0}^{\infty} X[n]z^{-n}$.

The input $A[n]$ in Fig. 4 is the same as $R_i[n]$ defined previously. The state variable $B_0[n]$ is the deviation of buffer occupancy from some desired buffer level B . The value of B could be set at any value between zero and B_{\max} , where B_{\max} is the buffer size. In practice, we may set it at $B_{\max}/2$. As a deviation from B , $B_0[n]$ can be positive or negative. Let C be the number of bits removed from the buffer for transmission per frame-period. Then $\Delta B_0[n] = B_0[n] - B_0[n-1] = R_o[n] - C$. Instead of defining $R_o[n]$ to be $f_n(D[n])$, we could add a constant offset and define $R_o[n]$ to be $f_n(D[n]) - C$ for all n . With this redefinition, which is adopted in Fig. 4, $\Delta B_0[n] = R_o[n]$. This redefinition of $R_o[n]$, however, does not necessitate a redefinition of $\Delta R[n] = R_o[n] - R_o[n-1] + \Delta r[n]$ (since the constant offset is canceled out), and the same encoder model as in Fig. 3(b) can be used for analysis. This also means that the same modified encoder model as in Fig. 3(c) can be used, and this is reflected in Fig. 4.

We now consider a solution to prevent unnecessary feedback control due to coding-mode switching. Instead of exercising feedback control based on $B_0[n]$, which will fluctuate naturally with the switching of coding modes, we use a filter with transfer function $H_1(z)$ to filter out this fluctuation before the feedback is computed. Let $B_1[n]$ be the filtered response on which the feedback control is based. To be robust against coding-mode switching, a criterion for the filter is that $B_1[n]$

must be constant if $\Delta B_0[n]$ is periodic in N , the period of the coding-mode cycle.

A simple $H_1(z)$ that could be used is $H_1(z) = (1/N)(1 + z^{-1} + \dots + z^{-(N-1)})$, which corresponds to averaging $B_0[n]$ over the most recent N frames. With this $H_1(z)$, the transfer function of the overall buffer-occupancy filter is given by

$$H_f(z) = \frac{1}{N} \left(\frac{1 + z^{-1} + \dots + z^{-(N-1)}}{1 - z^{-1}} \right). \quad (4)$$

We defer the investigation of the effects of $H_f(z)$ on the overall system to Section IV.

C. Feedback Compensator

There are many possible designs for the feedback compensator. This paper focuses on a linear proportional and derivative (PD) [13] controller. The value of $\Delta r[n+1]$ that will be given to the encoder in the next frame is

$$\Delta r[n+1] = \alpha_1 B_1[n] + \alpha_2 (B_1[n] - B_1[n-1]) \quad (5)$$

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

$$H_c(z) = \alpha_1 z^{-1} + \alpha_2 z^{-1} (1 - z^{-1}). \quad (6)$$

The first term is associated with DC compensation and the second term is for stabilizing the system.

Without the DC compensator, the buffer occupancy cannot be restored to the desired level B whenever there is a DC change in $A[n]$. That is, starting with the buffer level at B , a DC change will induce a change in $B_0[n]$, the deviation from B . As will be shown in Section IV, this deviation does not evolve back to zero 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 DC change is positive) or underflow (if the DC change is negative) should there be any further DC changes later.

The second term in (5) reacts more quickly to changes in $B_1[n]$ than the first term does. An increase in $B_1[n]$ means that the buffer occupancy is in the upward trend, and the encoder will be requested to lower its output bit rate through this term. As will be shown in Section IV, the system is unstable if $\alpha_2 = 0$ and is stable only for a certain range of α_2 .

IV. ANALYSIS

This section analyzes the overall system in Fig. 4. The analysis assumes that the state variables are not limited by the dynamic range of the underlying physical entities. In practice, 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 fluctuation, and this in turn helps us determine the buffer size required.

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

$$\begin{aligned}
\frac{B_1(z)}{A(z)} &= \frac{H_f(z)}{1 + H_f(z)H_c(z)/(1 - z^{-1})} \\
&= \frac{(1 - z^{-N})(1 - z^{-1})}{N(1 - z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\}(1 - z^{-N})} \\
\frac{B_0(z)}{A(z)} &= \frac{1}{(1 - z^{-1}) + H_f(z)H_c(z)} \\
&= \frac{N(1 - z^{-1})^2}{N(1 - z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\}(1 - z^{-N})} \\
\frac{\Delta r(z)}{A(z)} &= \frac{H_f(z)H_c(z)}{1 + H_f(z)H_c(z)/(1 - z^{-1})} \\
&= \frac{(1 - z^{-N})(1 - z^{-1})\{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\}}{N(1 - z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\}(1 - z^{-N})} \\
\frac{r(z)}{A(z)} &= \frac{H_f(z)H_c(z)}{(1 - z^{-1}) + H_f(z)H_c(z)} \\
&= \frac{(1 - z^{-N})\{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\}}{N(1 - z^{-1})^3 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\}(1 - z^{-N})}. \tag{7}
\end{aligned}$$

A. Stability Analysis

One goal of a feedback control system is to stabilize an unstable system. 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 rate of the encoder when it is too high and increase the rate when it is too low 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 [13]. For our adaptation controller, α_1 and α_2 must be within a certain range for stable operation. Stability is defined in the *bounded-input-bounded-output* sense here: given a bounded $A[n]$, all of the state variables should also be bounded.

It can be easily seen (by intuitive argument) that if $B_1[n]$ is bounded, then $B_0[n]$, $r[n]$, and $\Delta r[n]$ are also bounded. Thus, we focus only on the transfer function of $B_1[n]$. For stability, the poles of the transfer function must lie within the unit circle in the complex plane [13]. From (7), this means that the roots of

$$N(1 - z^{-1})^2 + \{\alpha_1 z^{-1} + \alpha_2 z^{-1}(1 - z^{-1})\} \frac{1 - z^{-N}}{1 - z^{-1}} = 0 \tag{8}$$

must have magnitudes less than one. Notice that we have divided the denominator in (7) by $(1 - z^{-1})$ to reflect that the factors $(1 - z^{-1})$ in the denominator and numerator cancel each other.

1) *Necessity of Derivative Feedback for Stabilization:* A necessary condition for stability is $\alpha_2 > 0$. To show this, apply the bilinear transformation $z = (s + 1)/(s - 1)$ to (8)—this transforms the interior (exterior) of the unit circle of the z -plane to the left (right) half of the s -plane, or, in other words, $|z| < 1$ if and only if $\text{Re}[s] < 0$. Using the Routh test [13], it is straightforward to show that the resulting equation has a root in the right half of the s -plane when $\alpha_2 = 0$ [14].

2) *Stable Regions for $N = 1$ and 2:* For $N = 1$ and 2, the bilinear transformation and Routh test can also be used to show that the stable regions are, respectively [14],

$$\begin{aligned}
4 - \alpha_1 - 2\alpha_2 &> 0 \\
8 - 2\alpha_1 - 4\alpha_2 - 4\alpha_1/\alpha_2 &> 0. \tag{9}
\end{aligned}$$

3) *Case of $\alpha_1 = 0$, General N :* The stable region when $\alpha_1 = 0$ is

$$0 < \alpha_2 < 2N \sin^2(\pi/2N). \tag{10}$$

The lower bound in (10) has been addressed. The upper bound is derived in the Appendix using a technique that traces the movement of the poles as α_2 is increased from zero. Note that (10) indicates that DC compensation, although desirable, is not necessary for stability.

4) *General Stable Region:* The general stable region for arbitrary $N \geq 1$ cannot be solved in closed form. In the Appendix, however, it is shown that the boundary of the region can be found by solving the following two equations numerically:

$$\begin{aligned}
-8N \cos(N\theta/2) \sin^3(\theta/2) + 2\alpha_1 \cos(\theta/2) \sin(N\theta/2) &= 0 \\
4N \sin^2(\theta/2) - \alpha_1 - 2\alpha_2 &= 0. \tag{11}
\end{aligned}$$

Fig. 5 shows the stable regions for $N = 2, 4, 8$, 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 .

B. Robustness against Coding-Mode Switching

The input signal component associated with coding-mode switching can be modeled by a periodic $A[n]$ with period N . This $A[n]$ would have frequency components at frequencies $0, 1/N, 2/N, 3/N, \dots$ (in unit of per-frame-time).

For robustness against coding-mode switching, we set the criterion that $\Delta r[n] = 0$ for any periodic input $A[n]$ with period N . Otherwise, there will be image-quality fluctuations caused by $\Delta r[n]$. In other words, if $\Delta r[n]$ is not zero, the system will hunt around for the right distortion level forever in a cyclic manner, even in the absence of significant scene changes.

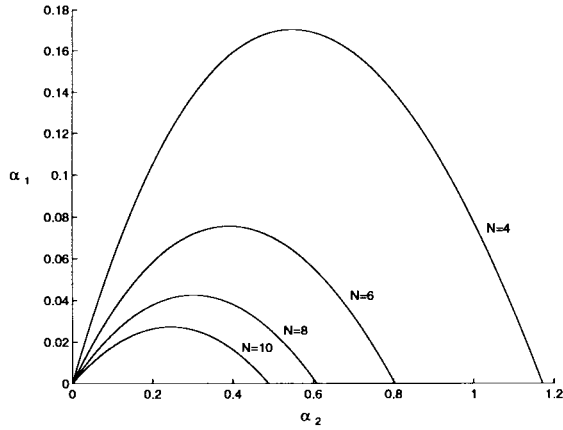


Fig. 5. The stable region for various N .

It is straightforward to show from (7) that, in general, the frequency response

$$\tilde{H}_{\Delta r}(f) = \left. \frac{\Delta r(z)}{A(z)} \right|_{z=e^{j2\pi f}}$$

is zero at $f = 0, 1/N, 2/N, \dots$. Thus, $\Delta r[n] = 0$ when $A[n]$ is periodic in N . This means that the system, if stable, will “zoom” in to a constant distortion level without hunting around forever.

C. Step Response and Sample Response

Recall from Section III-A that a scene change in a P or B frame can be simulated by a sample (impulse) input and that a long-term change in scene complexity can be simulated by a step input. In a linear time-invariant system, the unit-sample response can be obtained from the unit-step response by discrete-time “differentiation.” We shall therefore concentrate only on the unit-step response here.

Necessity of Proportional Feedback for DC Compensation: Suppose that initially all the state variables are at the zero state. A key question is whether the buffer deviation $B_0[n]$ will decay to zero after the initial rise due to a step input (see Section III-C). We now examine this issue. The z -transform of the unit-step response of $B_0[n]$ is $(B_0(z)/A(z)) \cdot (1/(1-z^{-1}))$. Applying the final-value theorem [13]

$$\begin{aligned} \lim_{n \rightarrow \infty} B_0[n] &= \lim_{z \rightarrow 1} (1-z^{-1}) \frac{B_0(z)}{A(z)} \frac{1}{1-z^{-1}} \\ &= \lim_{z \rightarrow 1} \frac{B_0(z)}{A(z)}. \end{aligned} \quad (12)$$

It can be shown from (7) that $\lim_{z \rightarrow 1} (B_0(z)/A(z))$ is zero if $\alpha_1 > 0$ and is nonzero if $\alpha_1 = 0$, proving that the proportional feedback is necessary for DC compensation of $B_0[n]$.

In general, the larger the value of α_1 , the faster is the decay of $B_0[n]$ to zero. Fig. 6 plots the unit-step responses of $B_0[n]$, $r[n]$, and $\Delta r[n]$ for $(\alpha_1, \alpha_2) = (0.003, 0.10)$ and $(0.009, 0.17)$ when $N = 10$. Comparatively, when $(\alpha_1, \alpha_2) = (0.003, 0.1)$, $B_0[n]$ rises to a higher peak and takes a longer time to decay to zero after that; $r[n]$, on the other hand, rises to a lower peak. Also, there are no oscillations in the decay pattern in

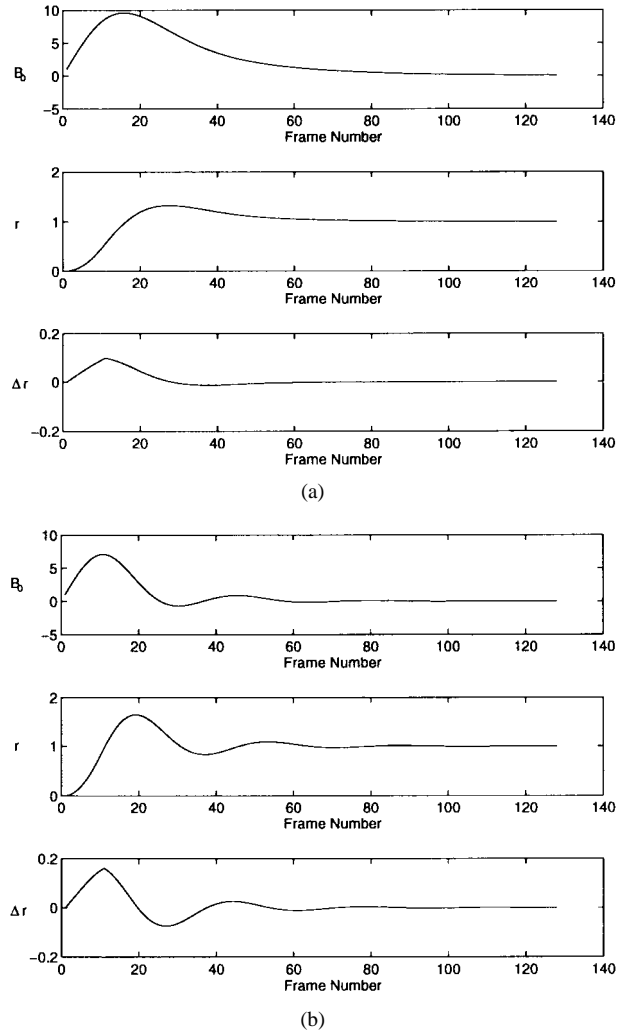


Fig. 6. Unit-step responses of the system. (a) $(\alpha_1, \alpha_2) = (0.003, 0.10)$. (b) $(\alpha_1, \alpha_2) = (0.009, 0.17)$.

the $(0.003, 0.10)$ case, but there are in the $(0.009, 0.17)$ case. The former is a typical weak-feedback setting and the latter a strong one. The latter is preferred if the buffer fluctuations (or delay jitters) are to be minimized and the buffer is small; the shortcoming, however, is that $r[n]$ overshoots by a larger amount before zooming in on the correct operating value of $r[n] = 1$. 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 equal the output rate, hence avoiding overflow and underflow. Also, note that the $r[n]$ overshoot in the transient behavior is inevitable if $B_0[n]$ is to eventually settle to zero, and this will be shown in Section V.

For a more comprehensive study of the effects of adjusting α_1 and α_2 on the unit-step responses, Fig. 7(a)–(c) plots three important parameters as functions of α_1 and α_2 within the stability region: 1) the peak $B_0[n]$, denoted by B_0^* ; 2) the peak overshoot of $r[n]$, denoted by $r^+ = r^* - 1$, where r^* is the peak of $r[n]$; and 3) the peak of $\Delta r[n]$, denoted by Δr^* . These graphs are useful design aids. For instance, Fig. 7(a) indicates that B_0^* decreases rapidly as α_1 and α_2 increase slightly from

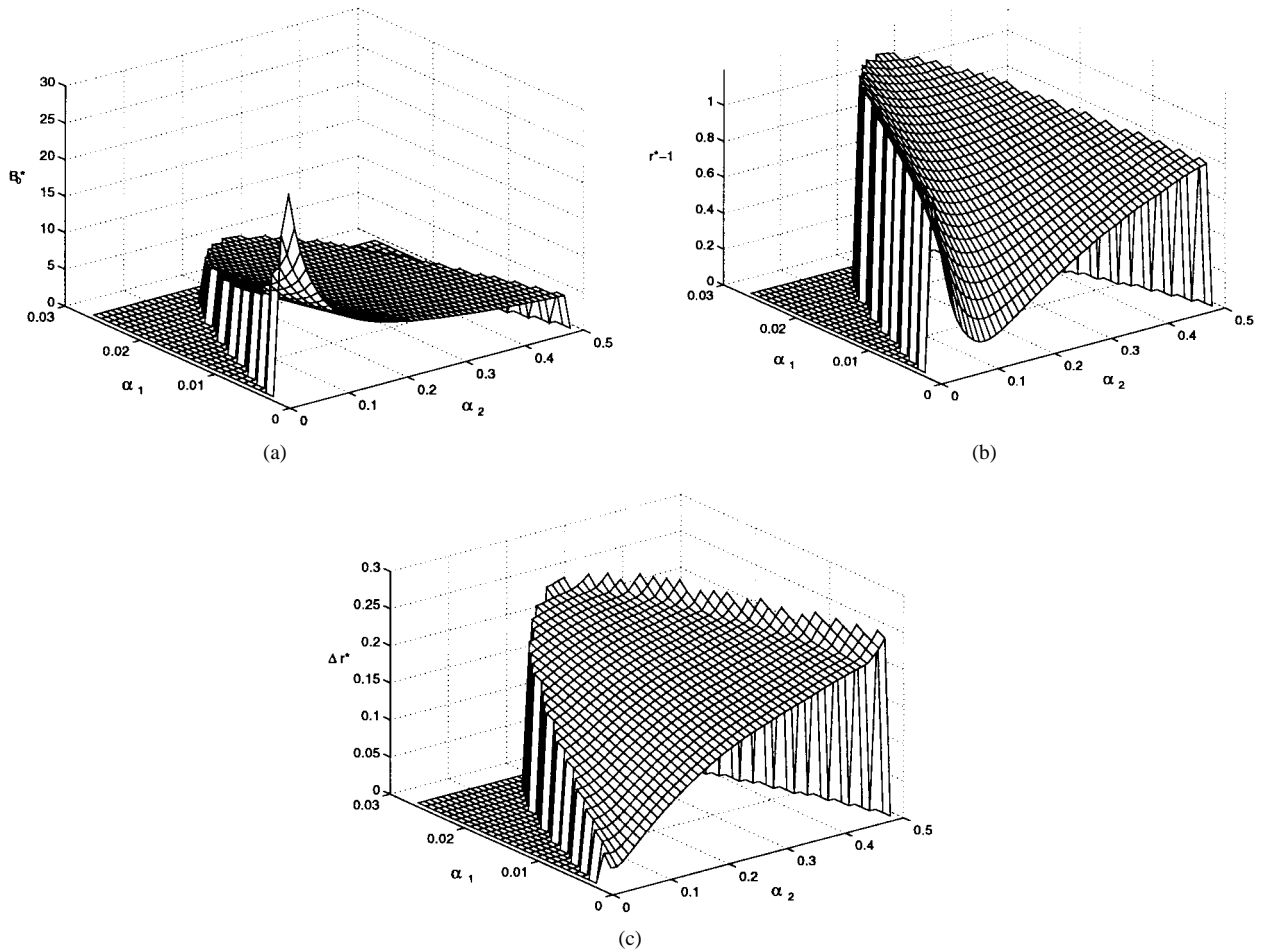


Fig. 7. (a) B_0^* , (b) $(b)r^+ = r^* - 1$, and (c) r^* as functions of α_1 and α_2 .

the zero points. After that, a further rise in the value of α_1 and α_2 does not decrease B_0^* by much. As shown in Fig. 7(b) and (c), however, the same rise in the value of α_1 and α_2 may increase r^+ and Δr^* in a nonnegligible way. This means that we should focus our design on small values of α_1 and α_2 .

V. TRADEOFFS BETWEEN IMAGE-QUALITY AND BUFFER-OCCUPANCY FLUCTUATIONS

This section presents a graphical approximation for analyzing the tradeoffs between image-quality and buffer-occupancy fluctuations, showing that there is a kind of “uncertainty principle” between them. Therefore, a designer of an adaptation controller cannot arbitrarily optimize one characteristic without undermining the other.

As a reference, we sketch in Fig. 8 the qualitative relationship among the unit-step and unit-sample responses of $\Delta r[n]$, $r[n]$, and $B_0[n]$. For notation purposes, we prime the unit-sample responses (in linear systems they are simply the discrete-time differentiation of the unit-step responses).

Let us first concentrate on the unit-step response. Assuming the initial conditions $r[-1] = 0$ and $B_0[-1] = 0$, for a unit-step input we have

$$\Delta B_0[n] = 1 - r[n], \quad n = 0, 1, \dots \quad (13)$$

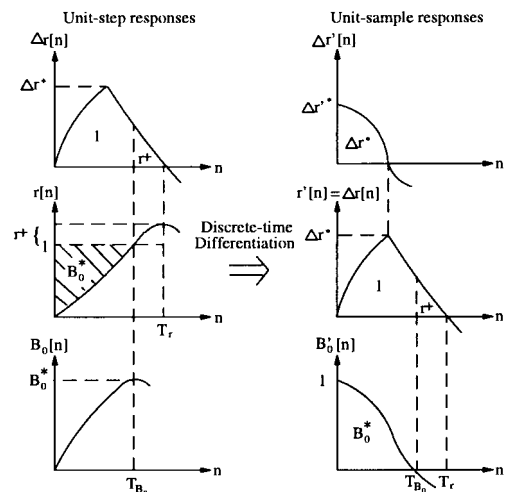


Fig. 8. Qualitative relationships among the unit-step and unit-sample responses of $\Delta r[n]$, $r[n]$, and $B_0[n]$.

Note that there is no loss of generality in the assumption of $r[-1] = 0$ since the superposition principle can be applied to understand the case where $r[-1] \neq 0$; in other words, the responses calculated here are triggered by a DC change, and they can be added to other responses to get the complete responses.

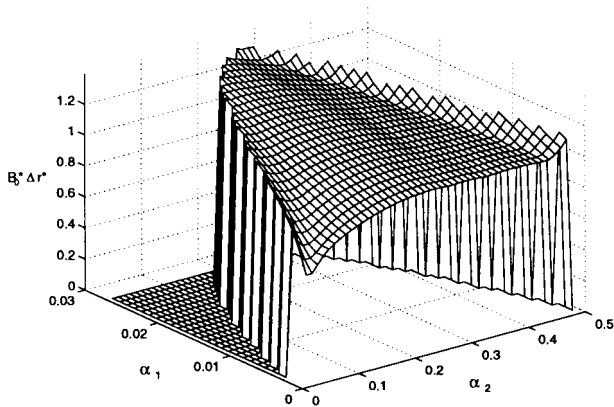


Fig. 9. Product of B_0^* and r^* for varying α_1 and α_2 .

With (13), given $B_0[n]$, we can derive $r[n]$, from which $\Delta r[n] = r[n] - r[n-1]$ can also be determined. The sketch in Fig. 8 takes these relationships into account. We assume that the system is stable and has been designed to have the desired characteristic that $B_0[n]$ peaks only once and then decays to zero. Let the time at which $B_0[n]$ peaks be denoted by T_{B_0} .

For proper operation, $r[n]$ should eventually settle to one after the initial transient response so that the buffer input rate is equal to the output rate. Before $r[n]$ settles to one, it overshoots to $(1 + r^+)$ at time T_r . The reason that this overshoot is unavoidable for a DC-compensated system can be seen from (13)—in order for $B_0[n] = \sum_{j=0}^n \Delta B_0[j] \rightarrow 0$, we must have $\sum_{j=0}^n r[j] \rightarrow n + 1$, or the average $r[j]$ over $n + 1$ frames must be one; but, for this to hold, $r[j]$ at some point must overshoot because $r[0] = 0$.

We shall use a continuous-time approximation here. From the curve of $r[n]$, applying triangular approximation on the shaded area, we have

$$B_0^* \approx T_{B_0}/2. \quad (14)$$

For the curve of $\Delta r[n]$, apply triangular approximation for the area under curve. We have

$$\Delta r^* \approx \frac{2(1 + r^+)}{T_r}. \quad (15)$$

From (14) and (15), we get

$$B_0^* \Delta r^* \approx \frac{T_{B_0}}{T_r} (1 + r^+). \quad (16)$$

If we approximate $r[n]$ to be a straight line passing through the three points $(0, 0)$, $(T_{B_0}, 1)$, and $(T_r, 1 + r^+)$, then the right-hand side (RHS) of (16) is one. This yields

$$B_0^* \Delta r^* \approx 1 \quad (17)$$

which is the uncertainty principle between B_0^* and Δr^* . For a system with $N = 10$, various combinations of α_1 and α_2 have been tested and Fig. 9 shows that in fact $B_0^* \Delta r^*$ is in the vicinity of one in the stable region.

To summarize, for a unit-step input, B_0^* is the peak buffer deviation and Δr^* corresponds to the maximum frame-to-frame image-quality adjustment, and (17) describes the trade-off between them. For linear systems, the unit-sample response is the differentiation of the unit-step response. This is

illustrated in the right column of Fig. 8. In particular, $r'[n] = \Delta r[n]$ and therefore Δr^* also corresponds to the maximum cumulative change in image quality given a unit-sample input before the cumulative change decays back to zero.

VI. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The abstract adaptation-controller model described in Section III can be implemented in many different ways, depending on how the encoder adjusts itself according to the feedback Δr . For experimentation purposes, we focus on a particular implementation in this section. Subsection A describes the implementation and Subsection B presents the experimental results.

A. Frequency-Plane Implementation

We assume that the video is compressed *a priori* by a standard MPEG encoder at very good image quality and stored. This encoder will be referred to as the preliminary encoder. The encoder in the adaptation controller is a secondary encoder. It takes a video sequence from the storage and adapts it for transport over a CBR channel. In doing so, it may need to further compress video in response to the feedback control by discarding data.

We assume the input to the secondary encoder to be MPEG-coded video, in which discrete-cosine transform (DCT) was performed on blocks of 8×8 pixels in each frame. The 64 spatial frequency components are ordered and, when dictated by the feedback, the secondary encoder will drop the data in each block starting from the highest frequency component to the lowest frequency component such that all blocks in the frame receive the same visual quality and the target bit allocation is met [2], [14]. Favoring lower frequency components over higher ones is a popular technique when the traffic generated by an encoder needs to be limited [2], [3]. Typically, some frequencies will be dropped and some retained to arrive at an effective distortion level for the frame.

Recall that a key element in our adaptation controller is the computation of the bits-versus-distortion functions of frames (see Section III-A). Let us refer to bits-versus-distortion functions as “profiles.” In our implementation, the profile of a frame n is obtained from the profiles of all of its constituent blocks by

$$f_n(D) = \sum_m B_{mn}(D) \quad (18)$$

where $B_{mn}(D)$ is the profile of block m in frame n . That is, for a given distortion level D the sum total of the bits required by the blocks is the number of bits required by the frame.

In our experiments, the distortion of a block (expressed in decibels) is defined as

$$\text{distortion of a block} = 10 \log_{10} \sum_l (s_l - s'_l)^2 \quad (19)$$

where s_l and s'_l are the values of pixel l within the block before and after the secondary compression, respectively. In practice, because a frequency component is either retained or dropped in its entirety, there is only a finite number of possible

distortion levels for each block. This means that $B_{mn}(D)$ of block m is a staircase function and it should be interpreted as the number of bits needed such that the distortion of that block does not exceed D .

At the frame level, distortion D means that all of the constituent blocks have distortion of no more than D .

After obtaining the profile of a frame n , the secondary encoder computes the new operating distortion level $D[n]$ according to the mechanism described in Section III-A. That is,

$$D[n] = f_n^{-1}\{f_n(D[n-1]) - \Delta r[n]\} \quad (20)$$

Given $D[n]$, the number of bits allocated to each block m is $B_{mn}(D[n])$, and frequency components are retained and dropped to meet this allocation.

B. Experimental Results

The video sequence used in our experiments is captured from the movie *Jurassic Park* and is 8 s in duration. The resolution and frame rate are 320×240 and 30 frames per second, respectively (i.e., a quarter of the size of the NTSC standard). The video sequence was previously coded at a good quality by a standard MPEG encoder with coding-mode switching period $N = 10$. Fig. 2(a) shows the video traffic before adaptation at the frame-level time scale. The mean and standard deviation of bits per frame taken over the 8-s duration are 35.3 and 20.3 kbits, respectively. In all experiments, the buffer size B_{\max} is set at $10 \times C$ bits (i.e., 10 frames). The frame-level signal-to-noise ratio (SNR), defined as follows, is used to measure the image quality of a frame:

$$\text{SNR} = 10 \log_{10} \frac{\sum_l s_l^2}{\sum_l (s_l - s'_l)^2} \quad (21)$$

where s_l (pre-MPEG compressed) is the original value of pixel l and s'_l is the pixel value after adaptation. The summations are over all pixels in the frame.

1) *Experiment 1—Smoothing with no Feedback:* Smoothing with no feedback corresponds to $\alpha_1 = \alpha_2 = 0$ in our model and the system is therefore unstable. When the CBR channel bandwidth C is set at the mean rate, 35.3 kb/frame, of the video sequence, buffer overflow occurs shortly after the DC of the video traffic starts to increase. When buffer overflow occurs, the high-frequency coefficients of the MPEG data are discarded. In Fig. 10, we plot the buffer occupancy as well as the SNR of the video sequence with this open-loop control. We see that the degradation of the SNR persists for some time until the buffer level drops again. In general, unless the CBR channel bandwidth C is equal to the video-traffic peak rate, buffer overflow may occur when the incoming traffic rate is higher than C for a sustained period of time.

In a separate experiment, we increased C until no buffer overflow occurs. We find that C must be 1.2 times the mean rate before buffer overflow can be prevented. It should be noted that this is only for an 8-s video clip. For a movie that lasts for more than one hour, there is generally a much larger fluctuation in bit rate, and it is expected that the capacity C needed to prevent buffer overflow will be much larger.

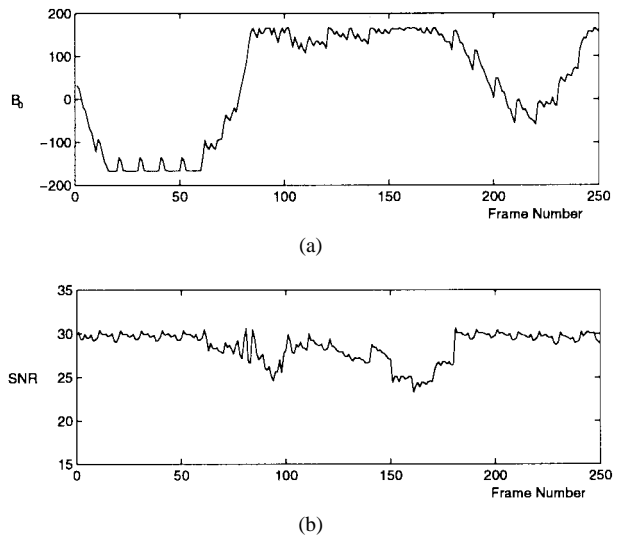


Fig. 10. (a) The buffer occupancy (in milliseconds) with no feedback control. (b) The corresponding SNR (in decibels).

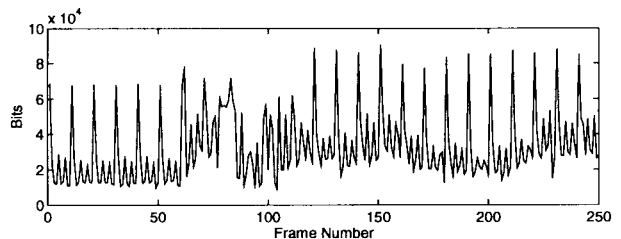


Fig. 11. The bit rate (in terms of bits per frame) of the sequence after adaptation.

2) *Experiment 2—Adaptation with Feedback Control:* In the next experiment, we fix $\alpha_1 = 0.003, \alpha_2 = 0.10$ and then $\alpha_1 = 0.009, \alpha_2 = 0.17$ for two different control settings. The desired buffer status B is set to $B_{\max}/2$, where B_{\max} is the buffer size.

We first present results related to $(\alpha_1, \alpha_2) = (0.003, 0.10)$. The traffic of the video sequence after adaptation is shown in Fig. 11. The control $\Delta r[n]$, the buffer-occupancy deviation $B_0[n]$ (normalized by C so that it is expressed in terms of milliseconds), and the SNR of the sequence after adaptation are shown in Fig. 12(a)–(c), respectively.

Note that the fluctuations in the bit rate of 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 average input-traffic rate is higher than C [e.g., frames 60–100, see Fig. 2(a)], the secondary encoder drops data in the manner described in Section III-A to restore $B_0[n]$. This results in 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.

With our implementation which interprets the encoder in the adaptation system as a secondary encoder (see Section III-A), there is a dynamic-range limit. That is, for a particular frame, the number of bits output by the secondary encoder cannot exceed the number of bits output by the preliminary

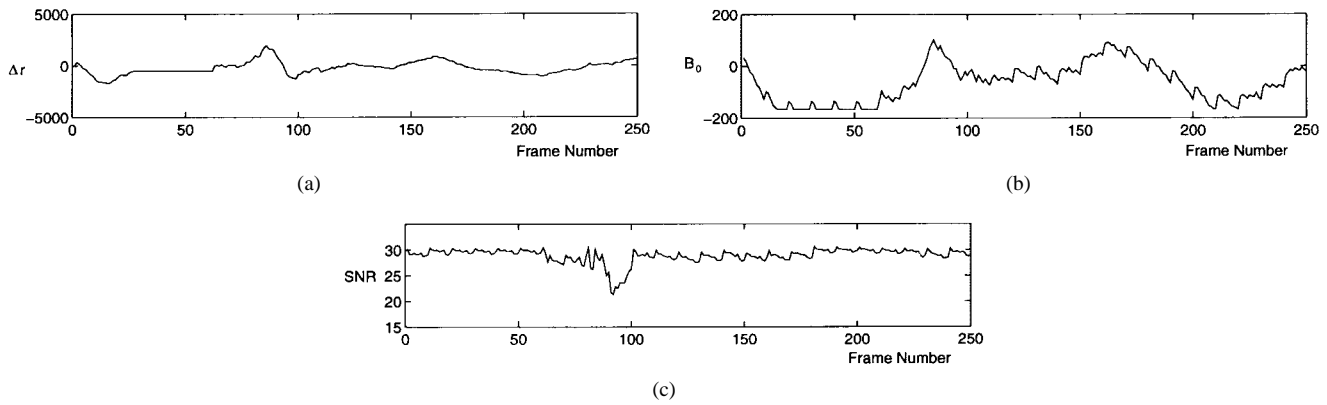


Fig. 12. Performance of the adaptation scheme with $\alpha_1 = 0.003$, $\alpha_2 = 0.1$, and $C =$ mean rate of the sequence. (a) The control $\Delta r[n]$ (in bits). (b) The buffer occupancy deviation $B_0[n]$ (in milliseconds). (c) The SNR (in decibels) of the output sequence.

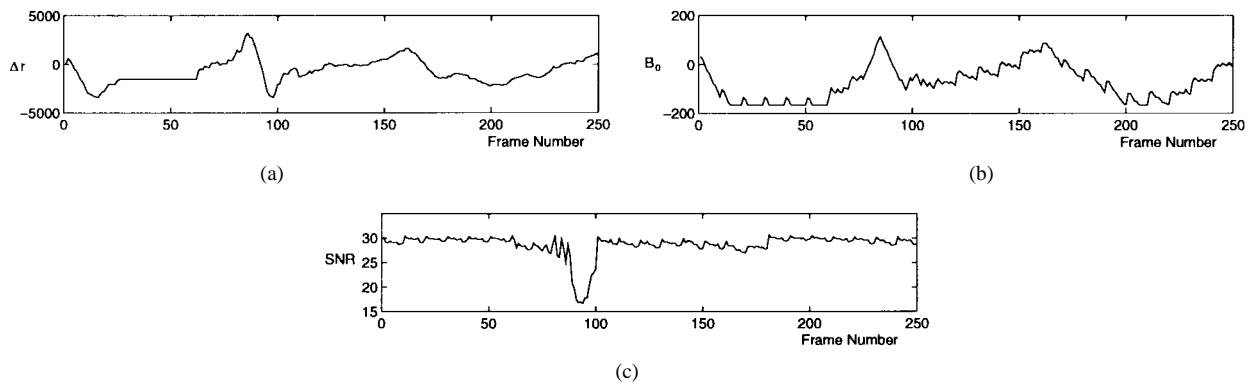


Fig. 13. Performance of the adaptation scheme with $\alpha_1 = 0.009$, $\alpha_2 = 0.17$, and $C =$ mean rate of the sequence. (a) The control $\Delta r[n]$ (in bits). (b) The buffer occupancy deviation $B_0[n]$ (in milliseconds). (c) The SNR (in decibels) of the output sequence.

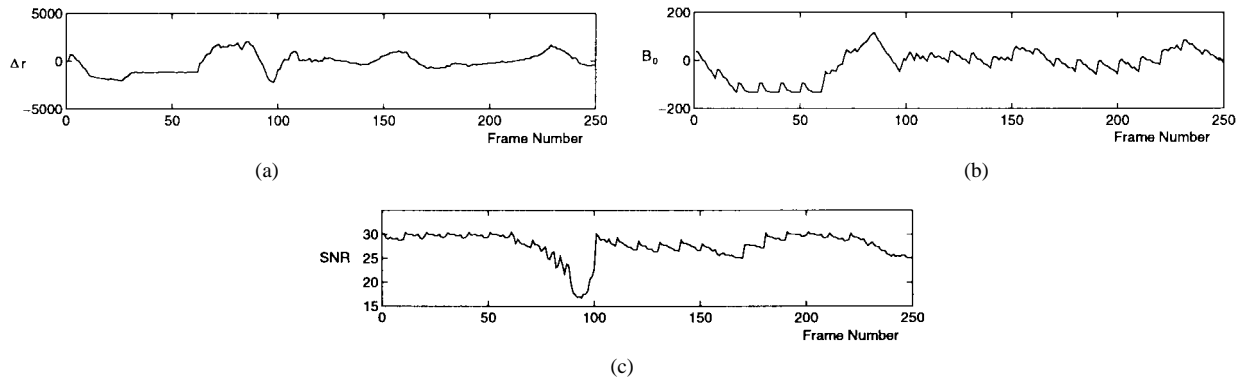


Fig. 14. Performance of the adaptation scheme with $\alpha_1 = 0.009$, $\alpha_2 = 0.17$, and $C = 0.8$ of the mean rate: (a) the control $\Delta r[n]$ (in bits), (b) the buffer occupancy deviation $B_0[n]$ (in milliseconds), and (c) the SNR (in decibels) of the output sequence.

encoder. In the experiment, C was set to the mean bit rate (35.3 kb/frame) of the video sequence. However, as indicated in Fig. 2(a), the initial output rate of the preliminary encoder is much lower than C . Therefore, B_0 soon reaches the underflow point even though all bits from the preliminary encoder are accepted. Although the feedback is a negative Δr instructing the secondary encoder to further improve the image quality,² the dynamic-range limit has been reached and, therefore, the

²Note that the Δr plotted in Fig. 12(a) is the actual bit adjustment. It differs from the intended bit adjustment when the dynamic-range limit is reached.

underflow situation is sustained until the output rate of the preliminary encoder starts to pick up.

We now compare the results of various settings of (α_1, α_2) . From the SNR plots in Fig. 10 (where $(\alpha_1, \alpha_2) = (0, 0)$) and Fig. 12 [where $(\alpha_1, \alpha_2) = (0.003, 0.10)$] we see that the closed-loop control can restore the SNR faster than the open-loop control can. To prevent buffer overflow in the closed-loop control, however, the worst-case SNR becomes slightly lower. The results for $(\alpha_1, \alpha_2) = (0.009, 0.17)$ are plotted in Fig. 13. Compared with when $(\alpha_1, \alpha_2) = (0.003, 0.10)$, its $\Delta r[n]$ is

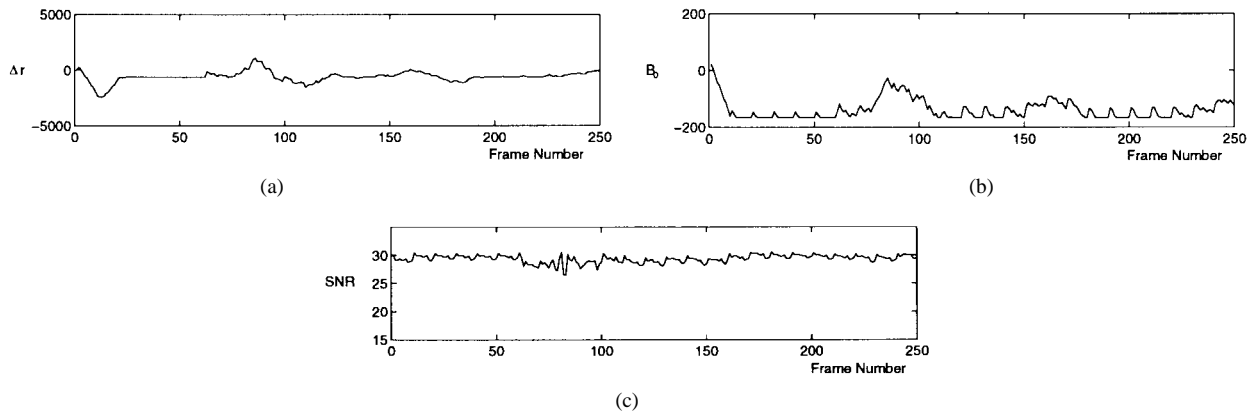


Fig. 15. Performance of the adaptation scheme with $\alpha_1 = 0.009$, $\alpha_2 = 0.17$, and $C = 1.2$ of the mean rate. (a) The control $\Delta r[n]$ (in bits). (b) The buffer occupancy deviation $B_0[n]$ (in milliseconds). (c) The SNR (in decibels) of the output sequence.

larger (especially for frames 60–100), and this corresponds to a less steady image quality. However, the buffer-occupancy deviation is restored more quickly. These qualitative tradeoffs confirm our analysis in the preceding two sections.

3) *Experiment 3—Varying Amount of Reserved Bandwidth:* In many situations, the mean bit rate of the video sequence is not known *a priori*, especially when the video is generated by a live capture (e.g., for video conference). Also, the mean bit rate may not be a useful engineering parameter for a long video sequence whose average rate (say over several seconds) may change drastically from time to time.

To verify that our adaptation scheme can prevent buffer overflow even when C is lower than the mean rate, we conducted an experiment in which C is set to 0.8 times the mean rate with the control setting $(\alpha_1, \alpha_2) = (0.009, 0.17)$. The experimental results are shown in Fig. 14. As can be seen, although the overall image quality of the video is worse than that when C is equal to the mean rate, both the image quality and buffer occupancy are still steady.

Finally, the results for C equal to 1.2 times the mean rate are plotted in Fig. 15. As can be seen, the SNR becomes rather smooth in this case.

VII. CONCLUSIONS

We have proposed and investigated in detail a control-theoretic framework for adapting VBR-compressed video for transmission on a CBR communications channel using a smoothing buffer. The scheme is based on a linear-feedback model and has the advantage of being readily analyzable compared with previous adaptation schemes. This allows systematic analysis and design of the adaptation controller to take into consideration issues such as stability, robustness against scene changes and coding-mode switching, and tradeoffs between image and buffer-occupancy fluctuations. An abstract feedback control model has also been realized with a specific implementation and experiments have been conducted to verify the performance of the controller as predicted by analysis.

This paper has assumed that the output channel is CBR with fixed bandwidth C . More generally, we could also consider scenarios in which the channel bandwidth $C[n]$ is a function

of time n and it varies according to the congestion level in the communications network. The congestion level, for instance, could be indicated to the adaptation controller in the form of a feedback from the network. The modeling of the adaptation controller remains unchanged. Instead of interpreting the input $\Delta B_0[n]$ in Section III-B (Fig. 4) as the input bits to the buffer minus C , we interpret it as the input bits minus $C[n]$. Step and sample responses can also be used to study the effects of a sudden change in $C[n]$. Thus, most of the results in this paper are still relevant.

Many areas for further research are possible. For instance, one could couple the output of the adaptation controller to network-access control that allows some variability in traffic rate. Reference [15] considers the case of open-loop adaptation (i.e., $\alpha_1 = \alpha_2 = 0$ with respect to our model) with the leaky-bucket access-control mechanism. For our feedback adaptation controller, the existence of a leaky bucket affords some variability in $C[n]$. In particular, in addition to controlling $\Delta r[n]$, the controller may also adjust $C[n]$, subject to the leaky-bucket specification. When the token bucket is not empty, then a sudden increase in the input rate to the controller can be accommodated by increasing $C[n]$. A drop in input rate can be dealt with by reducing $C[n]$ rather than decreasing the buffer-occupancy level. In this way, the buffer-occupancy variability should be smaller, resulting in smaller feedback, which in turn yields smoother image quality. The detailed dynamics and the tradeoffs among design objectives remain to be further investigated.

Instead of linear feedback, one could also consider nonlinear feedback that might improve the system performance. In particular, it would be interesting to explore schemes in which the feedback is made to be very strong when the buffer is about to overflow or underflow and to be very weak when the occupancy is near the desired level in a nonlinear fashion. One might also explore whether buffer-occupancy filters superior to that adopted in this paper are possible.

Finally, the implementation in Section VI is only one of many possibilities. Our abstract model can also accommodate implementations in which the encoding process is not divided into the preliminary and secondary phases. Also, instead of dropping high DCT frequency components in response to Δr ,

the adjustment of the quantization factor can be considered. The details of these other implementations and their relative merits are interesting areas for further study.

APPENDIX GENERAL STABLE REGION

This Appendix fills in the missing derivations in Section IV-A. Only a sketch is given and the reader is referred to [14] for details. We first study the restricted case when $\alpha_1 = 0$ before treating the general case.

1) *Restricted Case* ($\alpha_1 = 0$): Substituting $\alpha_1 = 0$ into (8), we have the following characteristic equation whose roots must be within the unit circle of the z -plane for stability:

$$N(1 - z^{-1})^2 + \alpha_2 z^{-1}(1 - z^{-N}) = 0. \quad (\text{A.1})$$

We apply the following argument.

- 1) When $\alpha_2 = 0$ there are $(N-1)$ roots at $z = 0$ and double roots at $z = 1$. Only one of the roots at one cancels with that of the numerator of the transfer function, and so the system is unstable.
- 2) 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 zero remain within the unit circle. To see what happens to the double roots at one, substitute $z = 1 + \epsilon$ into (A.1) and ignore the power of ϵ higher than the second power. We get $\epsilon^2 + \alpha_2\{\epsilon + (N-1)\epsilon^2/2\} = 0$, of which one root is zero and one root is a small negative number. This means one root of (A.1) 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 .
- 3) As α_2 increases, the next question is at what value of α_2 does one of those roots within the unit circle cross the unit circle? This is the upper bound on α_2 for stable operation.

For the last step, we apply the transformation $z = e^{j\theta}$ in (A.1) [after we have factored out the term $(1 - z^{-1})$ in (A.1)]. With some manipulation, we get

$$e^{j(N\theta/2)} = \alpha_2 \frac{\sin\left(\frac{N\theta}{2}\right)}{2N \sin^2\left(\frac{\theta}{2}\right)} j. \quad (\text{A.2})$$

The modulus of the left-hand side (LHS) is one, and applying this on the RHS yields

$$\alpha_2 = \left| \frac{2N \sin^2\left(\frac{\theta}{2}\right)}{\sin\left(\frac{N\theta}{2}\right)} \right|. \quad (\text{A.3})$$

The real part of RHS is zero, and applying this on the LHS yields

$$\theta = \pm \frac{\pi(1 + 4m)}{N}, \quad m = 0, 1, \dots \quad (\text{A.4})$$

Therefore, from (A.3) and (A.4)

$$\alpha_2 = \left| 2N \sin^2 \left\{ \frac{\pi}{2} \left(\frac{1 + 4m}{N} \right) \right\} \right|. \quad (\text{A.5})$$

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 $m = 0$

$$\alpha_2^* = 2N \sin^2(\pi/2N). \quad (\text{A.6})$$

2) *General Case*: When $\alpha_1 \geq 0$, the same type of argument as above can be applied. Instead of (A.1), the original characteristic (8) 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 the unit circle (guaranteed when $0 < \alpha_2 < \alpha_2^*$ by the previous proof). The root at one cancels with that of the numerator, and so the system is stable. The second step is to show that for infinitesimally small α_1 and $0 < \alpha_2 < \alpha_2^*$, the system remains stable (using the same technique as above). The third step is to derive, given a positive $\alpha_2 < \alpha_2^*$, the value of α_1 at which a root crosses the unit circle as α_1 is increased from zero. Following through these steps [14] will yield the two equations given in (11). The numerical method for finding the critical α_1 for a given positive $\alpha_2 < \alpha_2^*$ is to vary θ from zero to π/N in the two equations and solve for α_1 and α_2 for each θ .

ACKNOWLEDGMENT

The authors thank L. Ngai for generating some of the experimental data.

REFERENCES

- [1] N. Ohta, *Packet Video: Modeling and Signal Processing*. Norwood, MA: Artech House, 1994, p. 164.
- [2] C.-Y. Tse and S. C. Liew, "Video aggregation: An integrated video compression and multiplexing scheme for broadband networks," in *Proc. IEEE INFOCOM '95*, pp. 439-446.
- [3] P. Pancha and M. El Zarki, "MPEG coding for variable bit rate video transmission," *IEEE Commun. Mag.*, pp. 54-66, May 1994.
- [4] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. of the ACM*, vol. 34, pp. 47-58, Apr. 1991.
- [5] L. Wang, "Bit rate control for hybrid DPCM/DCT video codec," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, Oct. 1994, pp. 509-517.
- [6] 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. Consumer Electron.*, vol. 39, Aug. 1993, pp. 545-554.
- [7] C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. Image Processing*, vol. 2, pp. 50-59, Jan. 1993.
- [8] *MPEG-2 Test Model 5*, Document ISO/IEC JTC1/SC29/WG11/93-400, Test Model Editing Committee, Apr. 1993.
- [9] D. M. Cohen and D. P. Heyman, "Performance modeling of video teleconferencing in ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, Dec. 1993, pp. 408-419.
- [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, pp. 176-180, Apr. 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. Circuit Syst. Video Technol.*, vol. 4, pp. 121-128, Apr. 1994.
- [12] B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-based traffic modeling for performance evaluation of integrated networks," in *Proc. IEEE INFOCOM '92*, pp. 75-84.
- [13] B. C. Kuo, *Automatic Control Systems, Sixth Edition*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [14] C.-Y. Tse, "Adaptation of variable-bit-rate compressed video for transport over a constant-bit-rate communications channel in broadband

networks," M.Phil. thesis, Chinese Univ. Hong Kong, Shatin, Hong Kong, 1995.

- [15] P. Pancha and M. El Zarki, "Leaky bucket access control," in *Proc. IEEE INFOCOM'95*, pp. 796-803.



Soung C. Liew (S'84-M'87-SM'92) received the S.B., S.M., E.E., and Ph.D. degrees from Massachusetts Institute of Technology, Cambridge, in 1984, 1986, 1986, and 1988, respectively.

From 1984 to 1988, he was a Research Assistant in the Local Communication Networks Group, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, where he investigated fundamental design problems in high-capacity fiber-optic networks. From March 1988 to July 1993, he was with the Network Sys-

tems Research Laboratory, Bellcore, Morristown, NJ, where he engaged in fiber-optic and broad-band network research. He is currently with the Department of Information Engineering, Chinese University of Hong Kong, Shatin, Hong Kong, as a Professor. His current research interests include multimedia applications, video transport over networks, broad-band network control and management, and fast packet switching. He initiated and coordinated the first interuniversity ATM network testbed in Hong Kong. He is currently leading a project to explore how advanced multimedia and networking technologies can be used to improve the teaching and learning process.



Derek Chi-yin Tse (S'94-M'95) received the B.Sc. degree in physics and computer science and the M.Phil. degree in information engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 1993 and 1995, respectively.

He was with the Broad-band Communications Laboratory, Chinese University of Hong Kong, Shatin, Hong Kong, as a Research Assistant, working on various areas related to video transmission over broad-band networks. Since October 1997 he has been with Newbridge Networks (Asia), Ltd.,

Hong Kong, as a Network Consultant.