

Statistical multiplexing using MPEG-2 video encoders

by L. Böröczky
A. Y. Ngai
E. F. Westermann

This paper presents a system for statistical multiplexing of several compressed video programs using MPEG-2-compatible video encoders. We propose a new external joint rate control algorithm to dynamically distribute the channel bandwidth among the program encoders such that the video quality is approximately equal in all programs. In our algorithm, the bit rate of each encoder is updated on the basis of the relative complexities of the programs measured at boundaries of groups of pictures (GOPs) and whenever scene changes are detected. The proposed algorithm requires no external preprocessing of the input video sources. Furthermore, as compared with previous work in this area, our algorithm is not restricted to operate only with encoders having the same GOP structure. Thus, the GOP boundaries at the different encoders need not be synchronized. Bit rate changes take place only at GOP boundaries, allowing the encoders to operate at a constant bit rate within GOPs. Overall, this results in a piecewise variable bit rate compression for each of the encoders. We also describe a strategy for decreasing the reaction delay of the system for scene changes. Experimental results show that the proposed multiprogram video compression

system results in good picture quality with no external preprocessing, despite its relative simplicity.

1. Introduction

In typical broadcast systems, such as in direct broadcast satellite (DBS) applications, multiple video programs are encoded in parallel, and the digitally compressed bitstreams are multiplexed onto a single, constant bit rate channel. The simplest approach to this multiprogram encoding is to divide the available channel bandwidth equally among all programs. This method has the disadvantage that at any instant in time, the resulting quality of the video programs is uneven because of the different scene content of the programs and changes of scene content over time. The explanation for this result lies in the rate-distortion theory [1]. To achieve equal video quality (i.e., equal distortion) for all programs, the available channel bandwidth should be distributed unevenly among the programs, namely, in proportion to the information content (e.g., complexity) of each of the video sources. Thus, the objective of statistical multiplexing is to dynamically distribute the available channel bandwidth among the video programs in order to maximize the overall picture quality of the system. This is achieved by using a joint rate-control algorithm that guides the operation of the individual encoders based on a continuous monitoring of the scene content of each of the video sources.

©Copyright 1999 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/99/\$5.00 © 1999 IBM

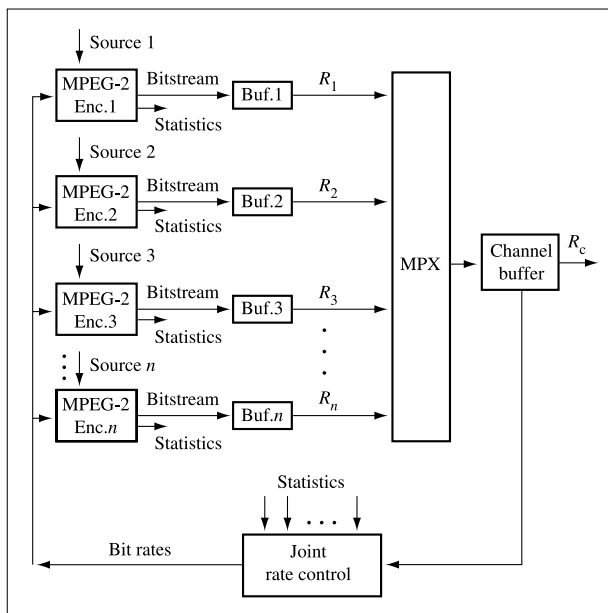


Figure 1

Multiprogram video compression system.

Basically, two different approaches can be distinguished for joint rate control: the feedback approach and the look-ahead approach. In the feedback approach [2–4], statistical measurements of video complexity are generated by the encoders as a by-product of the compression process. The statistics from all encoders are compared and used to control the bit allocation for the subsequent video. In the look-ahead approach [3, 5], the complexity statistics are computed by preprocessing all video programs prior to encoding. These statistics are then used to more accurately predict the bit rate allocation needed for optimum compression of the video sources in the rate-distortion sense. Finding the best statistics to describe the complexity of a program is a challenging task. In the feedback approach, the statistics are limited primarily to coding-related parameters. The look-ahead approach provides more freedom of choice, but at the price of extra computational complexity and additional cost. In either case, the main feature of the statistical multiplexing (stat-mux) system is that each encoder will produce a variable rate bitstream [6].

In this paper, we propose a solution for statistical multiplexing of MPEG-2 compressed video programs [7]. In particular, an external joint rate control algorithm is proposed that dynamically allocates bit rates for the program encoders using the feedback approach. In our algorithm, the bit rate for a given program encoder is updated only at boundaries of groups of pictures (GOPs), or

when a scene change is detected in the given program. This strategy allows the encoders to operate in a constant bit rate mode within the GOPs, resulting in piecewise-variable bit rate bitstreams. In contrast to previously published works in this area [2–6], the MPEG encoders in the proposed system are not required to have identical GOP structures. GOP boundaries may occur at arbitrary times in each encoded bitstream. Furthermore, for scene changes, a new GOP is started dynamically at or near the beginning of each new scene, ensuring quick reaction to video complexity changes. Because of these features, a channel buffer and a corresponding buffer control feedback loop are required in the proposed system.

In Section 2 we describe the proposed multiprogram video compression system. The joint rate control algorithm is presented in Section 3. The strategy for joint rate control in the event of scene change is described in Section 4. Determination of the minimum channel buffer size and the corresponding channel buffer control algorithm is given in Section 5. Finally, in Section 6, we present the experimental results obtained by computer simulations of the proposed system, followed by conclusions.

2. Multiprogram video compression system

Figure 1 shows an example multiprogram video compression system using the proposed feedback approach to joint rate control. The system consists of several MPEG-2 video encoders, buffers connected to each encoder, a joint rate controller, a multiplexer, and a channel buffer. The encoders produce bitstreams compatible with the MPEG-2 standard [7]. Along with the compressed bitstream, each encoder generates statistics related to the picture that has just been encoded. No preprocessing of the input sources is required, with the exception of scene change detection, which may be performed by the encoders or external to them.

The bit rate of each encoder is determined dynamically by the joint rate controller on the basis of the relative complexities of the programs and the occurrence of scene changes in the programs. Coding statistics generated by the encoders are input to the joint rate controller. The joint rate controller calculates the relative complexities of the programs and the bit rates based on these statistics. According to the proposed joint rate control algorithm, each encoder changes its bit rate only at GOP boundaries or near scene changes, where new GOPs are inserted. If a scene change does not occur, bit rate changes may still take effect at any GOP boundary. The reason for this is that the calculation of the program bit rates from GOP to GOP is based on the relative complexities of the programs. The joint rate controller acts to minimize the deviation of the sum of the program bit rates from the predefined channel bit rate. This scheme allows the encoders to operate at a constant bit rate (CBR) within

the GOPs using the CBR video buffer verifier model according to the MPEG-2 standard [6]. Overall, it results in a piecewise variable bit rate compression.

We emphasize that the encoders are not restricted to identical GOP structures and lengths. Since GOP boundaries in the encoders are not aligned in time and bit rates of each encoder are changed only at GOP boundaries, there are time intervals during which the sum of the individual bit rates is higher or lower than the predefined channel bit rate. To compensate for this occasional deviation from the channel bandwidth, a channel buffer is included in the system. Furthermore, feedback of channel buffer occupancy, or “fullness,” is incorporated into the joint rate control algorithm to prevent channel buffer overflow or underflow.

All MPEG-2 encoders used in the proposed multiprogram video compression system must be capable of providing at least the necessary coding statistics required by our joint rate control algorithm. In addition, encoders must have the ability to change bit rates at GOP boundaries. To further exploit the advantages of the proposed system in the event of scene changes, encoders must be able to change GOP structure dynamically, carry out scene change detection and reaction internally, or react to external scene change detection. As an example, IBM’s commercially available single chip MPEG-2 encoders fulfill the above requirements [8].

The following sections describe in more detail the proposed joint rate control algorithm, the required minimum channel buffer size, and the corresponding channel buffer control.

3. Joint rate control

The proposed joint rate control algorithm is based on the feedback concept. Statistics are produced by the encoders along with the compressed bitstream. These statistics are continuously fed into the joint rate controller from each encoder after compression of a picture. These coding statistics, together with the information on channel buffer fullness, are used to dynamically compute the bit rate allocation for the individual encoders. The bit rate of a program is proportional to the ratio between the complexity of that program and the sum of the complexities of all programs:

$$R_i = R_c \left(\frac{X_i}{\sum_i X_i} \right), \quad (1)$$

where R_i is the bit rate of program i , R_c is the channel rate, and X_i is the complexity of program i .

While other measures of video complexity are possible, in our algorithm the complexity of a picture is derived from the bit production model of MPEG-2 Test Model 5 [9]:

$$b_j = \frac{c_j}{Q_j}, \quad (2)$$

where the model parameter c_j is such that in order to produce a target number of bits b_j in picture j , the target quantization scale is set to Q_j . Using Equation (2), the bit rate of program i can be calculated for a GOP display time interval as

$$R_i = \frac{\sum_j (c_{ij}/Q_{ij})}{N_i/f_i}, \quad (3)$$

where c_{ij} is the bit production model parameter for picture j , Q_{ij} is the quantization parameter for picture j , N_i is the number of pictures in a GOP, and f_i is the frame rate of program i . In a stat-mux system, we wish to distribute the channel bandwidth among the programs such that

$$\sum_i R_i \leq R_c. \quad (4)$$

To achieve the goal of equalizing the picture quality of all programs, an ideal quantization parameter can be derived by using Equations (3) and (4):

$$Q_{\text{ideal}} = \frac{1}{R_c} \sum_i \left[(f_i/N_i) \sum_j c_{ij} \right]. \quad (5)$$

This ideal quantization parameter can result in equal picture quality for all pictures in each program. By substituting Q_{ideal} for Q_{ij} in Equation (3), the bit rate for a GOP in program i is calculated as

$$R_i = R_c \frac{(f_i/N_i) \sum_j c_{ij}}{\sum_i \left[(f_i/N_i) \sum_j c_{ij} \right]}. \quad (6)$$

In the proposed stat-mux system, c_{ij} is equal to $b_{ij}Q_{ij}$, where b_{ij} is a bit used for encoding picture j and Q_{ij} is the average quantization parameter in that picture. The complexity of a particular program is estimated as the average of the picture complexities over a sliding window of the GOP size of that program.

Equation (6) is used to determine dynamically the bit rates for each GOP of each encoder. As was explained previously, bit rate changes may occur in a program at any of the GOP boundaries, even if a scene change does not take place in that program. If bit rate changes are too abrupt in a program with no scene cut, the picture quality may vary significantly from GOP to GOP. Although the total quality of the system may improve, a noticeable change in picture quality between GOPs at the same scene is not desirable. To prevent this situation, a limit is placed on bit rate changes between GOPs of the same scene. In

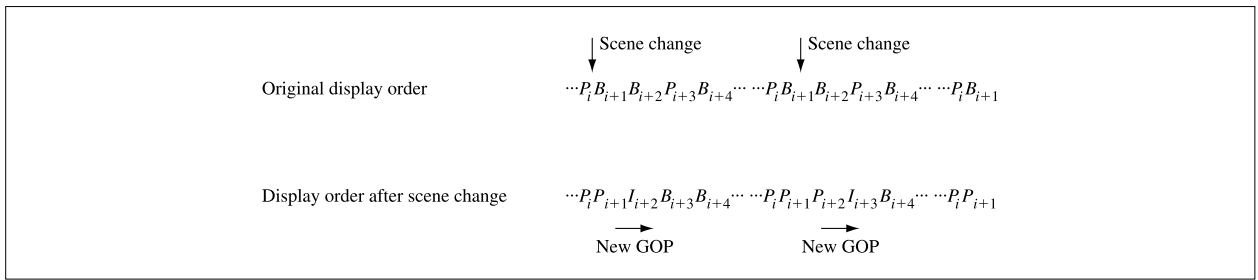


Figure 2

GOP structure changes at scene changes.

our experiments we allow a change of no more than 10% relative to the previous bit rate at the GOP boundary if no scene change occurs. If a scene change does occur, no limitation is placed on bit rate change.

4. Joint rate control at scene changes

In a program, scene changes may occur at any time. They may happen for any picture type and at any GOP position. If we assume that each encoder has its own fixed GOP structure and length and that bit rate changes are effective only at GOP boundaries, reaction of the system to complexity changes in the source programs may be slow because of the placement of the scene change within the GOP. To reduce the reaction time of the system to scene changes, the following strategy is set forth.

Let us assume that scene changes can be detected accurately either inside the encoders or externally, and that the location of the first picture of the new scene is known prior to the encoding of this first picture. Whenever a scene change is detected, the current GOP is ended prematurely. The first picture in the new scene is encoded as the last picture of the truncated GOP, because its statistics are used to predict the complexity of the new scene. These statistics are also used to calculate the bit rate for the first GOP of the new scene using Equation (6). This strategy allows a more insightful setting of the bit rate for the new scene, compared with depending upon default complexity values or average bit rate at the onset of the new GOP. **Figure 2** shows the original GOP structures and the new ones as scene changes occur. Three cases are distinguished by picture type at scene change occurrence.

The prediction of the new-scene complexity is based on the complexity of the first picture of the new scene and on empirically determined ratios among the complexities of the different picture types. If the picture type¹ of the first picture of the new scene (which is the last picture of the

truncated GOP) is P, every macroblock is encoded as an intra-macroblock, and the complexity is considered that of an I-picture. On the basis of this I-complexity, the average complexity of the new scene, X_i , is estimated as

$$X_i = \frac{X_I(1 + r_P n_P + r_B n_B)}{N_i / f_i}, \quad (7)$$

where X_I is the complexity of the I-picture, n_P and n_B are the number of P- and B-pictures in a GOP, and r_P and r_B are the ratios of the P- and B-picture complexities with respect to the I-picture complexity. Typical values of r_P and r_B are 0.5 and 0.25, respectively. The complexity X_i is used in Equation (6) for the bit-rate calculation of the new scene. As more pictures are encoded in the first GOP of the new scene, the complexity is continuously updated by applying the actual bit count and average quantization parameters used to encode the pictures.

Previously it was stated that the encoders are running in CBR mode inside the GOPs and that each encoder uses a CBR video buffer verifier model. No buffer underflow or overflow is allowed. Often, a goal of CBR rate control algorithms is to ensure that the buffer fullness at the end of a GOP is the same as the initial buffer fullness (e.g., 80% of the buffer size) prior to encoding the first picture of a sequence. However, this goal is not often achieved because of a mismatching of the target bit budget and the actual bits used per picture. Because of the overproduction or underproduction of bits in a GOP, the buffer fullness will be under or over the initial level at the end of the GOP, respectively. A considerable buffer fullness error may accumulate, resulting in a large bit surplus or deficit carried over to the next GOP. This rate control strategy works well if little or no bit rate change takes place at GOP boundaries. However, if abrupt bit rate changes do occur, a buffer fullness error (BFE) strategy is developed to further improve the picture quality at scene changes.

If a scene change is detected, the BFE is considered to be zero for the bit allocation of the first picture of the

¹ Picture types: P = predicted; I = intrapicture; B = bidirectionally predicted.

first GOP in the new scene. In this case, to prevent underflow or overflow of the encoder buffers, the bit rate calculated for this first GOP of the new scene must be modified by the BFE as

$$R_{i\text{mod}} = R_i + E(f_i/N_i), \quad (8)$$

where R_i is the calculated bit rate for program i according to Equation (6), E is the number of BFE bits to be eliminated, f_i is the frame rate for program i , and N_i is the number of pictures in a GOP. The bit rate of the program increases if the BFE is positive (the buffer fullness at the beginning of the GOP was less than the initial fullness at the start of encoding), or decreases if E is negative. This BFE strategy enhances overall picture quality at scene changes.

5. Channel buffer size and feedback control

Because the encoders can operate at different GOP lengths and structures, or may start to encode at different times, there may be time intervals when the sum of the individual bit rates is larger or smaller than the predefined channel bit rate. To remedy this, a channel buffer is required to output the multiplexed bitstream at exactly the channel bit rate. Two issues must be considered with respect to this buffer: One is the determination of the minimum size of the buffer, and the other is a control strategy to prevent channel buffer underflow and overflow.

Let us assume that the maximum total deviation of the sum of program bit rates from the channel bit rate is ΔR_{max} . In this calculation, it is valid to use the sum of the individual program bit rates because the bitstreams from each encoder are fed into each corresponding encoder buffer. These buffers output the bitstreams at exactly the calculated program bit rates, regardless of any bit rate fluctuations inside the GOPs. In the worst case, the maximum duration of this deviation can be as large as the longest GOP time among the encoders. For this case, the required minimum size of the channel buffer is determined as

$$B_s = 2\Delta R_{\text{max}} \text{tgop}_{\text{max}}, \quad (9)$$

where

$$\Delta R_{\text{max}} = \sum_i R_i - R_c$$

and tgop_{max} is the maximum GOP time. In Equation (9) a factor of 2 is used because both underproduction and overproduction of the channel bit rate are allowed. It is assumed that at first the buffer is filled to half of its size, $B_s/2$, after which it continuously outputs the multiplexed bitstream at the rate of R_c . In this case the time required to fill the buffer to half of its size represents the initial delay. As an example, using Equation (9), if the channel buffer output bit rate is 16 Mb/s, ΔR_{max} is 8 Mb/s, and

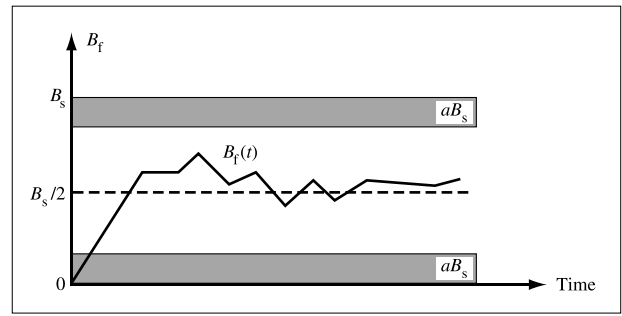


Figure 3

Channel buffer model.

tgop_{max} is 0.5 s, the minimum buffer size is 8 Mb. For this example, the corresponding initial delay is 0.25 s at a frame rate of 30 frames/s. Note that if a smaller channel buffer than the one determined by Equation (9) is desired for use in the stat-mux system, the maximum total deviation from the channel bit rate must be limited accordingly.

To prevent channel buffer underflow or overflow, the buffer model shown in Figure 3 is used. The channel buffer model includes predefined guard bands at the top and the bottom of the buffer. These guard bands are used to regulate the distribution of the bit rates. To prevent underflow and overflow, the buffer fullness B_f at any time must fulfill

$$0 < B_f(t) < B_s. \quad (10)$$

The parameter a determines the size of the guard band; for example, a may have a value of 0.25.

The three possible buffer fullness cases and their corresponding bit rate modifications are listed in the following subsections.

Case 1

Buffer fullness falls between the guard bands:

$$aB_s \leq B_f \leq (1-a)B_s.$$

In this case the calculated bit rates for the programs are generally not modified, except in extreme circumstances.

If $\sum R_i > R_c$ and $\sum R_i - R_c > (B_s - B_f)/\text{tgop}_{\text{max}}$,

$$\text{then } R_i = R_i \{ R_c + [(1-a)B_s - B_f]/\text{tgop}_{\text{max}} \} / \left(\sum R_i \right)$$

(no overflow);

if $\sum R_i < R_c$ and $R_c - \sum R_i > B_f/\text{tgop}_{\text{max}}$,

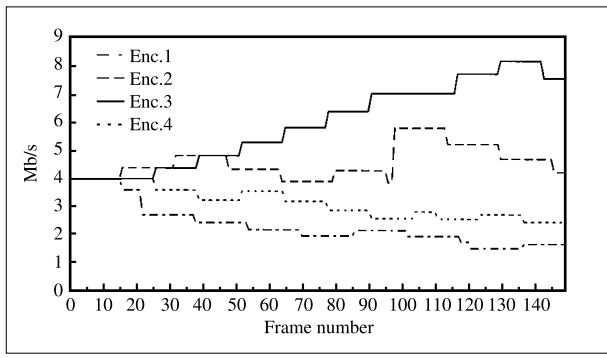


Figure 4

Program bit rates for encoding the first set of video sources at a channel bit rate of 16 Mb/s.

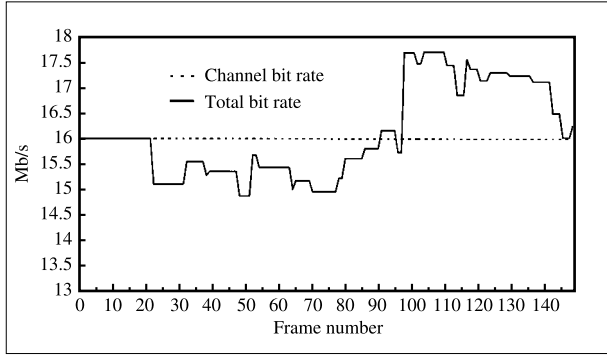


Figure 5

Total bit rate for encoding the first set of video sources ($R_c = 16$ Mb/s).

$$\text{then } R_i = R_i [R_c - (B_f - aB_s)/tgop_{\max}] / \left(\sum R_i \right)$$

(no underflow);

otherwise: no action.

Case 2

Buffer fullness falls in the upper guard band:

$$B_f > (1 - a)B_s.$$

In this case we allow only bit rate changes which will decrease the buffer fullness or maintain the current B_f .

If $\sum R_i > R_c$,

$$\text{then } R_i = R_i \left[R_c / \left(\sum R_i \right) \right]$$

(scaling down);

if $\sum R_i < R_c$ and $R_c - \sum R_i > B_f/tgop_{\max}$,

$$\text{then } R_i = R_i [R_c - (B_f - aB_s)/tgop_{\max}] / \left(\sum R_i \right)$$

(no underflow);

otherwise: no action.

Case 3

Buffer fullness falls in the lower guard band:

$$B_f < aB_s.$$

In this case we allow only bit rate changes which will increase the buffer fullness or maintain the current B_f .

If $\sum R_i < R_c$,

$$\text{then } R_i = R_i \left[R_c / \left(\sum R_i \right) \right]$$

(scaling up);

if $\sum R_i > R_c$ and $\sum R_i - R_c > (B_s - B_f)/tgop_{\max}$,

$$\text{then } R_i = R_i \{ R_c + [(1 - a)B_s - B_f]/tgop_{\max} \} / \left(\sum R_i \right)$$

(no overflow);

otherwise: no action.

6. Experimental results

To demonstrate the performance of the proposed system, several experiments were carried out, via simulation, using various image sequences and channel bit rates. We simulated the proposed multiprogram video compression system using four MPEG-2 encoders (Enc.1, Enc.2, Enc.3, and Enc.4). Each encoder had the capability of outputting the required coding statistics. Scene change detection was carried out inside the encoders. The video sources were chosen to represent widely differing scene contents and scene changes.

In our experiments, we have chosen relatively high channel bit rates (16–32 Mb/s), as our goal is to measure actual picture quality improvement achieved by our proposed system over that of fixed bit rate encoding. These higher bit rates allowed us to use nonfiltered, fairly complex, full D1 resolution input video sources, and enabled a more even visual comparison, especially at scene changes.

The first set of video sources were an *IBM Commercial*, *Table Tennis*, *Flower Garden & Mobile and Calendar (FG & MC)*, and a *Car* scene. The input frame rate was 29.97 frames/s with frame size of 720×480 pixels for each encoder. The sources were encoded in 4:2:0 chroma format. Two B-pictures were located between anchor

pictures. Closed GOP length was chosen as 16 in Enc.1 and Enc.2 and as 13 in Enc.3 and Enc.4. The channel rate was set at 16 Mb/s and the channel buffer size was 8 Mb, according to Equation (9). Each encoder began encoding at a bit rate of 4 Mb/s. This initial bit rate was changed dynamically according to the joint rate control algorithm.

Figure 4 shows the program bit rates allocated dynamically to each encoder using the proposed joint rate control algorithm. It can be seen that the *IBM Commercial* and the *Car* sequence had lower bit rates with respect to the other two sources.

Using the first set of video sources, the total bit rate, which is the sum of the program bit rates calculated dynamically for each of the four encoders, is given in **Figure 5**. The graph indicates the underproduction or overproduction of the channel bit rate, demonstrating the need for the channel buffer and for the feedback of its fullness to the joint rate controller.

The performance of the proposed system was compared with a scheme in which each encoder codes its source at a fixed bit rate (CBR encoding). The scene change detection was carried out by each encoder. **Table 1** shows the average peak signal-to-noise ratio (PSNR) values achieved by the proposed system and by CBR encoding at 4 Mb/s for the first set of video sequences. As the table indicates, the easy sources (*IBM Commercial*, *Car*) were encoded at a slightly lower quality in the proposed stat-mux system compared with encoding the sources by CBR at 4 Mb/s. However, this allows the proposed system to encode the more complex sources (*Table Tennis*, *FG & MC*) at a higher quality than the CBR model. The visual evaluation of the encoded sequences showed a better overall picture quality achieved by the proposed stat-mux system than the fixed bit rate model of the video sources.

We have also encoded the same set of video sources at a channel bit rate of 32 Mb/s with a channel buffer of 16 Mb. **Table 2** shows the average PSNR values achieved by the proposed system and by CBR encoding of each video source at 8 Mb/s.

To demonstrate the effectiveness of the channel buffer model and feedback control, **Figure 6** shows the channel buffer fullness during encoding of the sequences at a channel bit rate of 32 Mb/s. As the figure indicates, no channel buffer underflow or overflow occurred during encoding.

To illustrate the performance of the buffer fullness error (BFE) strategy, **Table 3** includes the PSNR values for the first pictures after scene changes using the proposed stat-mux system with and without the BFE strategy, at a channel bit rate of 16 Mb/s. As the table indicates, PSNR improvements of about 0.64–2.17 dB were achieved by using the BFE strategy, as compared with the system without it.

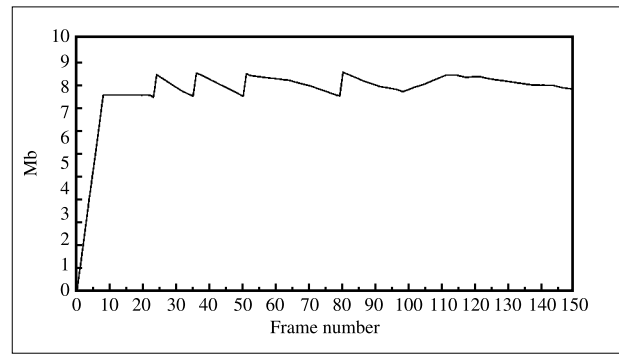


Figure 6

Channel buffer fullness during encoding of the first set of video sources at a channel bit rate of 32 Mb/s using a channel buffer of 16 Mb.

Table 1 Average PSNR values obtained by the proposed stat-mux system at a channel bit rate of 16 Mb/s vs. CBR encoding of each video source at 4 Mb/s.

Sources	Average PSNR (dB)	
	Stat-mux ($R_c = 16$ Mb/s)	CBR (4 Mb/s)
IBM Commercial (Enc.1)	38.48	40.11
Table Tennis (Enc.2)	32.11	31.29
FG & MC (Enc.3)	30.26	28.24
Car (Enc.4)	37.79	38.65

Table 2 Average PSNR values obtained by the proposed stat-mux system at a channel bit rate of 32 Mb/s vs. CBR encoding of each video source at 8 Mb/s.

Sources	Average PSNR (dB)	
	Stat-mux ($R_c = 32$ Mb/s)	CBR (8 Mb/s)
IBM Commercial (Enc.1)	40.49	42.40
Table Tennis (Enc.2)	35.36	34.61
FG & MC (Enc.3)	34.16	31.70
Car (Enc.4)	39.96	41.00

For the second set of experiments we used *IBM Commercial #2* (Enc.1), *Mixd* (Enc.2), *Football* (Enc.3), and *Mixe* (Enc.4) as input video sources. *Mixd* consists of the *Bike*, *Skyscrapers*, and *Basketball* sequences, while in *Mixe* the *Susie* sequence is followed by a *Forest with Cottage* scene. Because these sources are somewhat more complex than the first set of video sources, the channel bit rate was chosen as 24 Mb/s and the channel buffer was

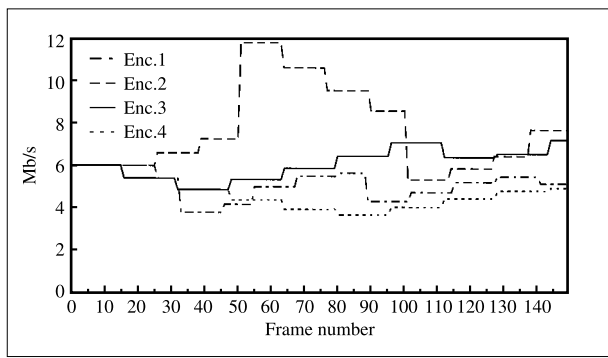


Figure 7

Program bit rates for encoding the second set of video sources at a channel bit rate of 24 Mb/s.

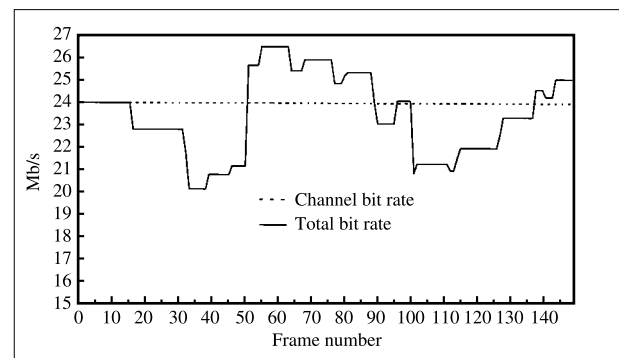


Figure 8

Total bit rate for encoding the second set of video sources ($R_c = 24$ Mb/s).

Table 3 PSNR values for the first pictures after scene changes using the proposed stat-mux system with and without the BFE strategy at a channel bit rate of 16 Mb/s.

Pictures	PSNR (dB)				PSNR (dB)			
	ENC.1				ENC.1			
	I ₂₂	B ₂₃	B ₂₄	P ₂₅	I ₁₂₁	B ₁₂₂	B ₁₂₃	P ₁₂₄
With BFE strategy	34.79	36.75	36.44	35.80	39.86	40.07	40.37	40.18
Without BFE strategy	34.15	36.09	35.77	35.09	38.72	39.08	39.30	38.99
Pictures	ENC.2				ENC.4			
	I ₉₈	B ₉₉	B ₁₀₀	P ₁₀₁	I ₁₁₁	B ₁₁₂	B ₁₁₃	P ₁₁₄
	I ₉₈	B ₉₉	B ₁₀₀	P ₁₀₁	I ₁₁₁	B ₁₁₂	B ₁₁₃	P ₁₁₄
With BFE strategy	30.49	33.52	33.49	32.55	39.00	39.09	38.48	38.77
Without BFE strategy	28.55	31.35	31.50	30.38	38.03	38.20	37.45	37.60

12 Mb. The coding parameters were identical to those of the first set of experiments, with the exception that the closed GOP length was 13 for Enc.1 and Enc.2, while it was 16 for Enc.3 and Enc.4. In the CBR case, the bit rate was fixed at 6 Mb/s. **Figure 7** shows the dynamic program bit rate changes for the encoders according to the joint rate control algorithm.

For encoding the second set of video sources, **Figure 8** shows the total bit rate as the sum of the calculated program bit rates, and the underproduction and overproduction of the channel bit rate during the encoding of the video sources.

The channel buffer fullness for encoding this second set of video sources is given in **Figure 9**. As this figure shows, there was no channel buffer underflow or overflow.

For the second set of video sources, **Table 4** includes the average PSNR values achieved by the proposed system and by CBR encoding at 6 Mb/s. This table, as well as the subjective evaluation of the encoded video sequences, showed the same trend in visual quality as was achieved for the first set of video sources. The stat-mux system resulted in a slightly lower video quality for easy sources (*IBM Commercial #2, Mixe*), while it improved the quality of the more complex image sequences (*Mixd, Football*) in comparison with CBR encoding.

7. Conclusion

A statistical multiplexing system for encoding multiple video programs in parallel using MPEG-2-compatible video encoders is proposed. The joint rate control

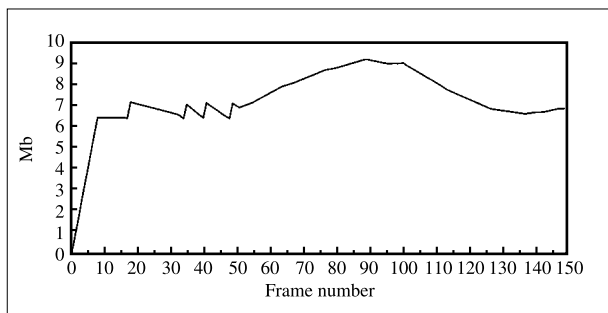


Figure 9

Channel buffer fullness during encoding of the second set of video sources at a channel bit rate of 24 Mb/s using a channel buffer of 12 Mb.

Table 4 Average PSNR values obtained by the proposed stat-mux system at a channel bit rate of 24 Mb/s vs. CBR encoding of each video source at 6 Mb/s.

Sources	Average PSNR (dB)	
	Stat-mux ($R_c = 24$ Mb/s)	CBR (6 Mb/s)
IBM Commercial #2 (Enc.1)	37.26	37.72
Mixd (Enc.2)	34.15	33.19
Football (Enc.3)	37.74	37.58
Mixe (Enc.4)	38.70	39.37

algorithm developed distributes the available channel bandwidth among the encoders on the basis of the relative complexities of the video sources and scene changes occurring within the programs. A special strategy has been developed to decrease the reaction delay of the algorithm for scene changes. This strategy results in a one-picture delay in reacting to scene changes and in enhanced picture quality of the new scene. The incorporated channel buffer and its feedback control into the joint rate controller allow the encoders to operate at various GOP lengths and structures and to begin encoding at different times.

The performance of the proposed system has been evaluated via simulation and compared with CBR encoding of video sources. Experimental results show that the developed multiprogram video compression system results in better overall picture quality with respect to the CBR model. This improvement is achieved without external preprocessing of the input video sources and in spite of the relative simplicity of the proposed system.

Acknowledgment

The authors wish to thank all who reviewed this paper, especially Dr. Peter Westerink from the IBM Thomas J. Watson Research Center, for their helpful comments.

References

1. T. Berger, *Rate Distortion Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.
2. G. Keesman, "Multi-Program Video Compression Using Joint Bit-Rate Control," *Philips J. Res.* **50**, No. 1/2, 21–45 (1996).
3. M. Perkins and D. Arnstein, "Statistical Multiplexing of Multiple MPEG-2 Video Programs in a Single Channel," *SMPTE J.* **104**, No. 9, 596–599 (1995).
4. L. Wang and A. Vincent, "Joint Rate Control for Multi-Program Video Coding," *IEEE Trans. Consumer Electron.* **42**, No. 3, 300–305 (1996).
5. A. Guha and D. J. Reininger, "Multichannel Joint Rate Control of VBR MPEG Encoded Video for DBS Applications," *IEEE Trans. Consumer Electron.* **40**, No. 3, 616–623 (1994).
6. M. Balakrishnan and R. Cohen, "Global Optimization of Multiplexed Video Encoders," *Proceedings of ICIP'97*, Santa Barbara, CA, October 26–29, 1997, Vol. 1, pp. 377–380.
7. "Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video," First Edition, *ISO/IEC 13818-2*, May 1996.
8. <http://www.chips.ibm.com/products/mpeg>.
9. "Test Model 5," *ISO/IEC JTC1/SC29/WG11 N0400*, April 1993.

Received November 30, 1998; accepted for publication May 26, 1999

Lilla Böröczky *IBM Research Division, Endicott, New York 13760 (boroczkl@us.ibm.com).* Dr. Böröczky is an Advisory Engineer/Scientist in the Encoder Development Department of the IBM Digital Video Products group. Her present responsibilities focus on video coding algorithms for MPEG-2 encoder products and their different applications. She received her M.Sc. in electrical engineering from the Technical University of Budapest, Hungary, in 1987 and her Ph.D. from the Delft University of Technology, Netherlands, in 1991. Before joining IBM in 1995, Dr. Böröczky was Senior Researcher at the KFKI Research Institute for Measurements and Computing Techniques of the Hungarian Academy of Sciences, Budapest, Hungary, and a Visiting Scholar at the Rensselaer Polytechnic Institute, Troy, New York. These previous assignments involved research on motion estimation, segmentation for digital video coding, and applications of image/video processing. She has authored or co-authored several papers on digital video processing and holds four patent applications in the field of digital video coding. Dr. Böröczky is a member of the IEEE and of the European Association for Signal Processing (EURASIP).

Agnes Y. Ngai *IBM Research Division, Endicott, New York 13760 (ngaia@us.ibm.com).* Ms. Ngai received a B.S. in electrical engineering from the City College of New York in 1973. She is a Senior Technical Staff Member at the IBM Endicott Center, working on digital video products, and is the chief architect on MPEG-2 video compression products. Joining IBM in 1973 at the development laboratory in Endicott, she has worked on the IBM S/370 system and on RISC processor development. Ms. Ngai holds several patents and has published several papers on MPEG video compression.

Edward F. Westermann *IBM Research Division, Endicott, New York 13760 (westerm@us.ibm.com).* Mr. Westermann joined IBM in 1979 in East Fishkill; he holds a B.S. in computer science from Union College, Schenectady, New York. He is an Advisory Software Engineer in the Encoder Development Department of the IBM Digital Video Products group. His current assignment is the design and coding of emulation software and behavioral modeling of the IBM MPEG-2 encoder chip set. Mr. Westermann has eleven patent applications in the field of digital video coding; he has co-authored several papers in this field. Previous assignments have included development of diagnostic and control software for IBM automated test equipment.