Submitted to Special Issue of Real-Time Imaging on Real-Time Digital Video over Multimedia Networks

Real-Time VBR Rate Control of MPEG Video Based upon Lexicographic Bit Allocation

Dzung T. Hoang

Sony Semiconductor Company of America

3300 Zanker Road, MS SJ3C3

San Jose, CA 95134

dzung@acm.org

Suggested Running Head: Real-Time VBR Rate Control of MPEG Video

Abstract

The MPEG-2 Video Standard describes in detail a bitstream syntax and a decoder model but leaves many details of the encoding process unspecified, such as encoder bit rate control. The standard defines a hypothetical decoder model, called the Video Buffering Verifier (VBV), that can operate in either constant-bit-rate (CBR) or variable-bit-rate (VBR) modes. In this paper, we present a low-complexity algorithm for VBR rate control suitable for low-delay, real-time applications. The algorithm is motivated by recent results in lexicographic optimal bit allocation. The basic algorithm switches between constant-quality and constant-bit-rate modes based on changes in the fullness of the VBV buffer. We show how the algorithm can be applied either to produce a desired quality level or to meet a global bit budget. Simulation results show that the algorithm compares favorably to the optimal VBR algorithm.

1 Introduction

Digital video has been steadily gaining momentum and acceptance in the marketplace since the adoption of video compression standards such as H.263 [1], MPEG-1 [2], and MPEG-2 [3]. Notable among the applications of digital video are the Digital Video Disk (DVD) and the Advanced Television Systems Committee (ATSC) Digital Television Standard [4]. Both employ MPEG-2 as the video coding technology.

The MPEG-2 Video Standard describes in detail a bitstream syntax and a decoder model but leaves many details of the encoding process unspecified. One area where this is the case is rate control, a process in which the bit rate of compressed pictures is regulated to satisfy constraints imposed by the transmission channel (or storage device) and buffers at the encoder and decoder. The standard defines a hypothetical decoder model, called the Video Buffering Verifier (VBV), that consists of a buffer, a decoder, and a display device, as shown in Figure 1. The VBV places quantifiable limits on variations in the bit rate of the encoded bitstream and is an integral part of the standard. One of the new features of the MPEG-2 Video Standard is support for variable-bit-rate (VBR) operation of the VBV in addition to the constant-bit-rate (CBR) mode found in MPEG-1.

With CBR encoding, the bit rate of the compressed bitstream must be controlled so that, on average, a constant rate is achieved. The instantaneous bit rate may differ from the constant rate since the output of the encoder is buffered before transmission. However, because the buffer is of finite size, the bit rate cannot vary too much for too long, otherwise the buffer would overflow or underflow. A major drawback to CBR encoding is that a constant bit rate does not necessarily translate to constant quality in the reconstructed video. The relationship between bit rate and quality is highly dependent on the content of the scene to be compressed. For example, an “easy” scene containing an image with few textures and edges and with little motion can be compressed at a relatively low bit rate with good quality. Conversely, a “difficult” scene with lots of detail and motion would need to be coded at a higher bit rate to achieve comparable quality. Because of the dependence of bit rate on the scene content, the quality of a CBR-encoded bitstream could vary significantly when there are many scenes of varying complexities, unless the bit rate is so high as to cause no visible artifacts for even the most complex scene.

VBR rate control allows video to be coded at rates that vary depending on the complexity of the scene, up to some maximum bit rate. An often-mentioned benefit of VBR is that video can potentially be coded at a constant visual quality. Furthermore, VBR can be more efficient than CBR for coding video with a desired minimum quality level. For the same level of quality, a VBR encoding could potentially use fewer bits than a CBR encoding since the bit rate in the CBR encoding would have to be tailored to code the most difficult scene with the same quality used in the VBR encoding. With CBR, less difficult scenes would be coded at a higher quality than needed, therefore wasting bits. With VBR, the desired quality can often be delivered with no wasted bits. VBR encoding is also useful for multiplexing multiple video bitstreams over a common CBR channel [5]. Assuming that the instantaneous bit rates of the video sequences are statistically independent, more VBR bitstreams with a given peak rate R can be multiplexed onto the channel than CBR bitstreams coded at a constant rate of R.

CBR rate-control algorithms such as MPEG Test Model 5 (TM5) [6] process video in a single pass and is suitable for low-delay, real-time encoding. By using adaptive bit allocation and quantization techniques, the rate control is able to adjust to changing scene complexities. For high-quality encoding applications that can operate an offline environment, where more computational resources and time are available, a multi-pass strategy would be more appropriate in order to achieve the best possible quality.

In this paper, we address VBR encoding in a real-time environment requiring low encoding delay. Applications fitting this requirements include consumer digital video cameras, digital VCR, and video telephony. Specifically, we present a new low-complexity VBR rate control algorithm that is based upon recently published results on optimal bit allocation under a lexicographic minimax framework [7]. The framework assumes knowledge of the rate-distortion characteristics of every input picture and requires offline processing, where the rate-distortion characteristics are first measured and modeled before a global bit allocation can be computed. With the new real-time rate control algorithm, we remove the need for offline processing while maintaining some of the advantages of the optimal algorithm.

This paper is organized as follows. We begin by reviewing the lexicographic bit allocation framework of [7] in Section 2. Based on the optimal VBR bit allocation algorithm, we then describe our low-complexity algorithm for VBR rate control in Section 3. We present simulation results in Section 4.

2 Overview of Lexicographic Bit Allocation

The lexicographic bit allocation framework is based upon three components: 1) a novel lexicographic optimality criterion, 2) a set of buffering constraints, and 3) a rate-distortion model of input pictures. Analysis of CBR and VBR bit allocation problems under this framework yields necessary and sufficient conditions for optimality that is related to the buffering constraints in a simple and intuitive manner. In this section, we review the underlying assumptions of the framework and summarize the optimality results.

2.1 Lexicographic Optimality Criterion

At the heart of the lexicographic framework is the lexicographic optimality criterion. The criterion states that an optimal bit allocation results in an encoding whose maximum distortion (measured on some basic unit such as a picture) is the least among all admissible allocations. Furthermore, the second highest distortion should also be the least among those admissible allocations whose maximum distortion is lowest, and so on. This is a refinement of the well-known minimax criterion. The term "lexicographic" refers to the observation that the definition of the criterion resembles the sorting of alphabetic strings. The analogy relies upon associating with each admissible allocation a string that consists of the distortion values sorted in decreasing order, so that the first element of the string is the highest distortion value. The allocation whose corresponding string is first in sorted order (lexicographically minimum) is optimal.

2.2 VBV Buffering Constraints

If we apply the lexicographic criterion to budget-constrained bit allocation without buffering constraints, then an allocation that results in constant distortion would be optimal. The lexicographic criterion comes into play when buffering constraints do not admit the trivial constant-distortion allocation. Buffering constraints arising from the MPEG-2 VBV are derived and used in the analysis presented in [7]. We now briefly describe the operation of the MPEG-2 VBV and present the buffering constraints.

2.2.1 CBR Operation

In CBR mode, the VBV takes in data from the channel at a constant rate. The data is stored in the VBV buffer before being processed by the decoder. At regular display intervals, the decoder retrieves all the bits needed to decode the next picture *instantaneously* from the buffer. Before retrieving bits for the first picture, the decoder waits until the buffer fullness reaches a specified level. The channel rate R, the VBV buffer size B_{V}, and the initial buffer fullness B_1 are parameters encoded in the compressed video bitstream. A compliant MPEG-2 bitstream must be decodable by the VBV model without causing the buffer to overflow or underflow.

Given a bit allocation $\mathbf{s} = \langle s_1, s_2, \ldots, s_N \rangle$ to the N pictures in a video sequence, the fullness of the VBV buffer can be described by the recurrence

$$\begin{aligned} B_{\mathrm{f}}(\mathbf{s}, 1) &= B_1, \\ B_{\mathrm{f}}(\mathbf{s}, n+1) &= B_{\mathrm{f}}(\mathbf{s}, n) + B_{\mathrm{a}}(n) - s_n, \end{aligned} \tag{1}$$

where $B_{\mathrm{f}}(\mathbf{s}, n)$ denotes the fullness of the VBV buffer just *before* the nth picture is removed from the buffer, T_n the amount of time required to display picture n, and $B_{\mathrm{a}}(n) = RT_n$ the number of bits that enter the buffer in the time it takes to display picture n.

To insure that the VBV buffer neither overflows nor underflows, an allocation $\mathbf{s}$ must satisfy the following for all n:

$$\max\{B_{\mathrm{f}}(\mathbf{s}, n) + B_{\mathrm{a}}(n) - B_{\mathrm{V}}, 0\} \le s_n \le B_{\mathrm{f}}(\mathbf{s}, n). \tag{2}$$

2.2.2 VBR Operation

The MPEG-2 Video Standard defines two VBR modes of operation. In the first mode, the **vbv_delay** parameter, which indicates to the decoder the VBV buffer fullness at which to start decoding the next picture, is set to a constant value of 65,535. In this mode, the buffer is filled at the peak rate R encoded in the bitstream until the buffer is full. When the buffer is full, the input bitstream is halted, but not discarded. This mode is suited for modeling a decoder connected to a storage device that can send the encoded bitstream on demand at a rate up to R.

In the second VBR mode, bits enter the VBV buffer at a piecewise-constant rate up to the peak rate R. The instantaneous transmission rate for a picture is determined by the coded **vbv_delay** parameter and the number of bits in the encoding of the picture. This mode is suited for modeling a channel with a variable delivery rate.

Only the first mode is analyzed in [7]. We restrict our attention to this mode and note that this is the mode used by the DVD-Video Standard.

Operation of the VBV buffer in the first VBR mode is described by the recurrence

$$\begin{aligned} B_{\mathrm{f}}(\mathbf{s},1) &= B_{\mathrm{V}}, \\ B_{\mathrm{f}}(\mathbf{s},n+1) &= \min\{B_{\mathrm{V}}, B_{\mathrm{f}}(\mathbf{s},n) + B_{\mathrm{a}}(n) - s_n\}. \end{aligned} \tag{3}$$

The minimization in (3) prevents the buffer from overflowing. When $B_{\mathrm{f}}(\mathbf{s},n) + B_{\mathrm{a}}(n) - s_n > B_{\mathrm{V}}$, we say that picture n results in a *virtual overflow*. The buffer may underflow if the encoding rate exceeds the peak rate for an extended period. To prevent this, an allocation $\mathbf{s}$ must satisfy the following for all n:

$$s_n \leq B_{\mathrm{f}}(\mathbf{s},n). \tag{4}$$

2.3 Rate-Distortion Modeling

In order to define rigorously an optimal allocation and to prove interesting results, the lexicographic framework relies on an explicit rate-distortion model for each picture. To facilitate implementation, the rate-distortion model takes the form of a rate-quantization function that relates the number of bits produced with a given quantization setting. Furthermore, the quantization scheme is assumed to be perceptually-tuned to the Human Visual System, in that pictures coded with the same perceptually-adjusted *nominal* quantizer should be of the same visual quality. Analysis in [7] assumes rate-quantization models that are monotonic and independent, in that the rate-quantization model of one picture is independent of the coding of another. Although this assumption does not hold in general for MPEG-2 encoders that employ motion-compensated prediction, simulations gave good results when steps were taken to correct for deviations that arise from coding dependencies.

2.4 CBR Optimality Results

With the three components of the framework in place, analysis of the buffer-constrained allocation problem for CBR constraints results in the following theorem that completely characterizes an optimal allocation.

Theorem 1 *For CBR buffering constraints, an admissible allocation* $\mathbf{s} = \langle s_1, s_2, \ldots, s_N \rangle$ *that uses nominal quantizers* $\mathrm{Q} = \langle Q_1, Q_2, \ldots, Q_N \rangle$ *is optimal if and only if the following conditions hold. Also, the optimal allocation is unique.*

1. *If* $Q_j > Q_{j+1}$ *for some* $1 \leq j < N$*, then* $B_{\mathrm{f}}(\mathbf{s}, j) = s_j$.
2. *If* $Q_j < Q_{j+1}$ *for some* $1 \leq j < N$*, then* $B_{\mathrm{f}}(\mathbf{s}, j+1) = B_{\mathrm{V}}$.

The theorem simply states that the nominal quantizers used in the optimal allocation only changes from one picture to the next when the buffer is either full or empty, increasing when the buffer is full and decreasing when the buffer is empty. The theorem can be understood by considering the complexity of successive scenes, where we define a scene as a segment of consecutive pictures that are coded with the same nominal quantization. The relationship between nominal quantization and average rate can be thought of as a measure of the encoding complexity of a scene. More complex scenes require higher nominal quantizers than scenes of less complexity, for the same average rate. Taking advantage of the available buffering, we can code segments of pictures with an average rate higher or lower than the channel rate R for a limited time. To minimize the maximum distortion, equivalently nominal quantizer, we should code a *locally* complex scene at a higher rate than the channel rate. To offset for the increased rate, adjacent scenes of lower complexity should be coded at lower rate than the channel rate. This is achieved by starting a complex scene with the buffer full and ending it with the buffer empty.

The theorem can also be interpreted as specifying how bits can be shifted locally from scenes of differing complexity. Bits can be shifted from a less complex scene to an adjacent more complex one within limit of the buffering constraints.

2.5 VBR Optimality Results

VBR mode differs from CBR in that there is no buffer overflow condition, thus allowing segments of video to be coded at an average rate lower than the peak for an extended time. In typical applications, the local average rate is lower than the peak for a large portion of the sequence. For example, an often-quoted average video bit rate for DVD-Video is 3.5 Mbps, whereas the peak video bit rate is 9.8 Mbps. Intuitively, to equalize quality most of the video should be coded at a constant quality at less than the peak rate. Complex scenes should be coded at a lower quality at the peak rate.

Analysis of the lexicographic framework results in the following theorem[1] which supports the above intuition.

Theorem 2 *For VBR buffering constraints, an admissible allocation* $\mathbf{s}$ *that uses nominal quantizers* $\mathrm{Q} = \langle Q_1, Q_2, \ldots, Q_N \rangle$ *is optimal for a given bit budget if and only if the following conditions hold. Also, the optimal allocation is unique.*

1. *If* $B_{\mathrm{f}}(\mathbf{s}, j) + B_{\mathrm{a}}(j) - s_j > B_{\mathrm{V}}$ *for* $1 \leq j \leq N$*, then* $Q_j = \min_{1 \leq k \leq N} Q_k$.

2. *If* $B_{\mathrm{f}}(\mathbf{s}, N) > s_N$ *then* $Q_N = \min_{1 \leq k \leq N} Q_k$.

3. *If* $Q_j > Q_{j+1}$ *for* $1 \leq j < N$*, then* $B_{\mathrm{f}}(\mathbf{s}, j) = s_j$.

4. *If* $Q_j < Q_{j+1}$ *for* $1 \leq j < N$*, then* $B_{\mathrm{f}}(\mathbf{s}, j+1) = B_{\mathrm{V}}$ *and* $B_{\mathrm{f}}(\mathbf{s}, j+1) + B_{\mathrm{a}}(j+1) - s_{j+1} \leq B_{\mathrm{V}}$.

The first two conditions establish a globally minimum nominal quantization that, in a sense, defines the base quality for the sequence. A picture that results in a virtual overflow should be coded at the base quality. The third and fourth conditions are similar to those in Theorem 1. Changes in nominal quantization, therefore, can only occur when the VBV buffer is either full or empty. In particular, this means that for a segment of pictures to be coded at higher than the base quantization, it must either start at the beginning of the sequence or with the buffer full. To return to coding at the base quantization, the buffer must empty. This means that a segment coded at higher than the base quantization must empty the buffer at some point or terminate the sequence. This implies that such a segment uses the maximum

[1]The version of this theorem found in [7] is incomplete. The version presented here comes from [8].

capacity of the VBV buffer. Informally, we call the segments that are coded at the base quality "easy" and the others "hard."

3 Real-Time VBR Rate-Control

In this section, we develop a real-time VBR rate control algorithm based upon the optimal results described above. We first outline the optimal VBR algorithm presented in [7] and then show how to modify this algorithm to work in a single pass.

3.1 Optimal Algorithm

Below is an iterative algorithm for computing an optimal VBR allocation that uses a total of B_{T} bits.

1. Mark all pictures as *easy*. Let $B_{\mathrm{easy}} \leftarrow B_{\mathrm{T}}$.
2. Allocate B_{easy} bits to easy pictures using a constant nominal quantizer. Let $Q_{\min}$ by the nominal quantizer used.
3. Simulate the VBV to identify *hard* segments of pictures. A hard segment leads to a buffer underflow when $Q_{\min}$ is used and consists of pictures that follow the most recent virtual overflow up to and including the picture the caused the underflow. After identifying a hard segment, reduce the bit allocation to the picture that caused the underflow to just prevent underflow. Reset the buffer fullness to empty and continue the simulation, adding new pictures to the existing hard segment if the buffer continues to underflow.
4. Let B_{hard} be the total number of bits allocated to the hard pictures. Let $B_{\mathrm{easy}} \leftarrow B_{\mathrm{T}} - B_{\mathrm{hard}}$.
5. If a new hard segment has been identified in Step 3, goto Step 2.
6. Allocate bits to maximal segments of hard pictures using an optimal CBR algorithm.

We make the following observations about the algorithm. The algorithm loops when a new hard segment has been detected. Since the number of hard segments is bounded by the length N of the sequence, the algorithm terminates after at most N iterations. With each iteration, $Q_{\min}$ is reduced. Halting the algorithm at the end of an iteration results in an optimal allocation for the bit budget used. In effect, additional iterations only refine the allocation to meet the original bit budget B_{T}.

3.2 Single-Pass Algorithm

To transform the above algorithm to operate in a single pass, we need to remove any looping and offline processing. This means that we can perform only one iteration. Also, we cannot perform Step 2 which assumes knowledge of the rate-quantization characteristics of the entire sequence to be able to compute the value required for $Q_{\min}$. The identification of hard pictures in Step 3 requires looking into the future to determine whether the current picture belongs to a hard segment. Finally, the optimal CBR algorithm in Step 6 is an offline algorithm and needs to be replaced with an online algorithm.

The design of efficient algorithms for online CBR rate control has been taken up by many researchers, for example [9, 10, 11, 12]. We do not propose yet another CBR algorithm, but show how to modify an existing one to use in our VBR algorithm.

The limitation of a single iteration and the lack of knowledge of rate-quantization characteristics impact the bit budget and the choice of $Q_{\min}$. One approach is not to impose a bit budget and set $Q_{\min}$ based on the desired quality. This would result in a final bit budget (and average bit rate) that depends on the complexity of the input sequence. Another approach is to carefully monitor and control the average bit rate to meet a specified goal. The approach to use would depend on the particular application. For example, storage applications that are capacity limited would require the latter approach, while quality-conscious applications would favor the former. We first describe an algorithm to deliver video at a desired base quality and later show how to modify it to meet a desired bit budget.

3.2.1 Basic VBR Algorithm

The single-pass VBR algorithm below removes the offline processing identified above and is suitable for low-complexity encoding with a specified base quality.

1. Initially the VBV buffer is set to full. In addition to the VBV buffer size B_{V}, peak input bit rate R,

and base quantization scale $Q_{\min}$, the algorithm has three additional parameters: a CBR trigger threshold t_{CBR}, a VBR trigger threshold t_{VBR}, and a target buffer fullness B_{tgt}, with $B_{\mathrm{V}} > t_{\mathrm{VBR}} \geq t_{\mathrm{CBR}} > B_{\mathrm{tgt}} > 0$.

2. The encoder simulates operation of the VBV and keeps track of the VBV buffer fullness B_{f}. The encoder operates in VBR mode and codes the input pictures using $Q_{\min}$ until $B_{\mathrm{f}} \leq t_{\mathrm{CBR}}$. With this event, the encoder switches to CBR mode and allocates bits to some number K of future pictures so that the target fullness B_{tgt} would be reached after the next K pictures have been encoded. The parameter K may correspond to small number of GOP periods, for example. In the CBR mode, the peak rate R is used.

3. A CBR rate-control algorithm, such as the one specified in TM5, can be used in CBR mode with one modification: the nominal quantization scale used to code any picture cannot be lower than $Q_{\min}$. The VBV buffer is to be operated near the target fullness B_{tgt}.

4. The encoder switches from CBR mode to VBR mode when $B_{\mathrm{f}} > t_{\mathrm{VBR}}$, when the base quantizer $Q_{\min}$ is again used.

The operation of the algorithm is demonstrated with a plot of buffer fullness Figure 2. In the VBR region above t_{VBR}, the algorithm operates in a constant-quality VBR mode. The CBR region below t_{CBR} marks where the algorithm operates in CBR mode at the peak input rate. The transition region provides hysteresis for transitions between VBR and CBR modes.

The single-pass VBR algorithm does not attempt to model the complexity of the coded pictures. Instead, it *reacts* to changes in complexity of pictures that are manifested as changes in the VBV buffer level. One implication of this reactive nature is a delay in switching between VBR and CBR modes compared to the optimal algorithm. Whereas the optimal algorithm switches to CBR mode when the buffer is full, the single-pass algorithm must wait until the buffer reaches t_{CBR} before switching. This leaves less buffering for the CBR-coded pictures. Another difference is that for segments with short spikes in complexity, the single-pass algorithm may switch to CBR coding when the optimal algorithm would continue to operate in VBR mode. The switching from CBR mode to VBR mode is less problematic. Because the CBR algorithm in Step 3 enforces a minimum nominal quantization of $Q_{\min}$, the switching to VBR mode can actually take place before the buffer reaches t_{VBR}. In fact, it is this enforcement of the minimum nominal quantization that enables the buffer to fill up as the complexity of coded pictures decreases.

3.2.2 VBR Algorithm with Bit Budget

As presented above, the single-pass algorithm does not attempt to control the coding rate to meet a specified bit budget. In a sense, enforcing a bit budget is akin to specifying a CBR constraint with the rate averaged across the entire sequence. The difference between CBR and VBR can be attributed to the size of a virtual encoding buffer. With CBR, the encoder buffer is of a fixed size, whereas with VBR the buffer's size is essentially unbounded. We can use this intuition to design a VBR rate control scheme that has a bit budget constraint.

In a sense, meeting a specified bit budget is a long-term process, whereas the basic VBR algorithm above can be viewed as a short-term process. This is similar to the approach taken in [13]. We propose to adapt an existing CBR rate control algorithm to perform the long-term rate control. For simplicity, we describe the process using the simple buffer-feedback rate control scheme used in TM5, where $Q_{\min}$ is computed directly from the fullness of a virtual encoder buffer. The size of this buffer determines the how quickly $Q_{\min}$ can change and also affects the accuracy of the rate control. Conceptually, the size of the virtual buffer need not be constant. For example, to come close to the specified bit budget, we may want to reduce the size of the buffer gradually to constrain variance in the aggregate rate near the end of the sequence. The virtual buffer should empty at a constant rate determined by the bit budget. In addition, we may want to reduce the rate by a small amount related to the buffer size to guarantee that the specified bit budget will not be exceeded.

4 Simulation Results

We implemented the VBR rate control algorithms using the ISO MPEG-2 software encoder [14] and provide simulation results with the same 3,660 frame sequence used in [7] for comparison.

As in [7], we use the TM5 adaptive quantization scheme to provide a mapping between nominal quantization and actual quantization scale. The nominal quantization scale is modulated by an *activity factor* that is computed from the spatial variance of the luminance blocks within a macroblock. In TM5, the actual quantization scale used for coding a particular macroblock is determined from an initially computed reference quantization scale, a feedback factor that is dependent of the state of a virtual encoding buffer, and the normalized activity factor for that block. In reporting results, we define the nominal quantization for a picture as the average of the product of the reference quantization scale and the buffer-feedback factor over all coded macroblocks.

We report results here for the following four algorithms: 1) optimal lexicographic VBR, 2) TM5 CBR, 3) single-pass VBR without long-term rate control, and 4) single-pass VBR with long-term rate control. The simulation results for the lexicographic VBR algorithm are taken directly from [7]. To speed up the simulations of the other algorithms, we use approximate motion estimation. This results in a small decrease in the quality of the TM5 results presented here compared to those in [7]. For both single-pass VBR algorithms, the quantization parameters were adjusted to give roughly the same encoded file size as TM5 and the following thresholds were used: $t_{\mathrm{VBR}} = 0.85B_{\mathrm{V}}$, $t_{\mathrm{CBR}} = 0.7B_{\mathrm{V}}$ and $B_{\mathrm{tgt}} = 0.3B_{\mathrm{V}}$. For the single-pass algorithm with long-term rate control, the size of the virtual encoder buffer was set to $20B_{\mathrm{V}}$.

The input sequence consists of 3,660 interlaced frames sampled at 720×480 resolution and 29.97 frames/sec. For all simulations, we used the GOP pattern `BBIBBPBBPBBPBBP` and a VBV size of 1,835,008 bits. For the VBR algorithms, we specified an average rate of 3.0 Mbps and a peak rate of 4.5 Mbps. For the TM5 CBR algorithm, we specified a constant rate of 3.0 Mbps.

A summary of the simulation results is listed in Table 1. The buffer fullness, nominal Q, and PSNR are plotted in Figures 3, 4, and 5, respectively. The results show that the VBR algorithms produce less variance in PSNR and nominal quantization compared to TM5 CBR. However, the average PSNR is actually better for TM5 CBR than for the VBR algorithms. From Figures 4 and 5, we observe that TM5 gives better PSNR for scenes in the first half of the sequence where the the VBR algorithms operate in constant-quality mode and worse PSNR on for complex scenes in the second half. Since the majority of the pictures operate in constant-quality mode, the average PSNR is biased towards TM5. Visually however, the VBR encodings have more even quality and better scene transitions. Between the optimal VBR algorithm and the single-pass algorithm without long-term rate control, the major differences can be attributed to the use of online versus offline CBR algorithms. The single-pass VBR algorithm with rate control shows some similarity with TM5 CBR in long-term variations in nominal quantization, but with smoother local variations in nominal quantization and PSNR.

5 Conclusion

We have presented a low-complexity VBR rate control algorithm suitable for low-delay, real-time encoding applications. The development of the algorithm is motivated by recent results in optimal lexicographic bit allocation. The algorithm is best suited to provide a desired quality level and can be used for short-term rate control. It can also be applied with a suitable long-term rate control strategy to meet bit budget constraints. Results from a challenging encoding simulation shows that the new algorithm retains advantages of the more complex optimal algorithm in equalizing quality, especially for scene transitions.

6 Acknowledgments

This work was performed when the author was employed by Digital Video Systems, Inc., 160 Knowles Drive, Los Gatos, California.

References

[1] ITU-T Study Group 15, Draft recommendation H.263 (Video coding for narrow telecommunication channels), April 26 1995, Document LBC-95-163.

[2] ISO, CD11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbits/s, Nov. 1991.

[3] ISO-IEC/JTC1/SC29/WG11/N0802, Generic coding of moving pictures and associated audio information: Video, Nov. 1994, MPEG Draft Recommendation ITU-T H.262, ISO/IEC 13818-2.

[4] Advanced Television Systems Committee, ATSC digital television standard, Sept. 1995, ATSC Doc. A/53.

[5] B. G. Haskell and A. R. Reibman, Multiplexing of variable rate encoded streams, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 417–424, Aug. 1994.

[6] ISO-IEC/JTC1/SC29/WG11/N0400, Test model 5, Apr. 1993, Document AVC-491b, Version 2.

[7] D. T. Hoang, E. L. Linzer, and J. S. Vitter, Lexicographic bit allocation for MPEG video, *Journal of Visual Communication and Image Representation*, vol. 8, no. 4, pp. 384–404, Dec. 1997.

[8] D. T. Hoang, *Fast and Efficient Algorithms for Text and Video Compression*, PhD thesis, Brown University, May 1997.

[9] E. Viscito and C. Gonzales, A video compression algorithm with adaptive bit allocation and quantization, in *SPIE Proceedings: Visual Communications and Image Processing*, Nov. 1991, vol. 1605, pp. 58–72.

[10] K.-W. Chow and B. Liu, Complexity based rate control for MPEG encoder, in *Proceedings ICIP'94*, Austin, TX, Nov. 1994, vol. 1, pp. 263–267.

[11] J. Choi and D. Park, A stable feedback control of the buffer state using the controlled lagrange multiplier method, *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 546–557, Sept. 1994.

[12] D. Park and K. Kim, Buffered rate-distortion control of MPEG compressed video channel for DBS applications, in *Proceedings IEEE Internation Conference on Communications*, 1995, vol. 3, pp. 1751–1755.

[13] W. Ding, Joint encoder and channel rate control of VBR video over ATM networks, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 266–278, Apr. 1997.

[14] MPEG Software Simulation Group, MPEG-2 encoder/decoder version 1.2, July 19, 1996, Available URL: `http://www.mpeg.org/MSSG`.

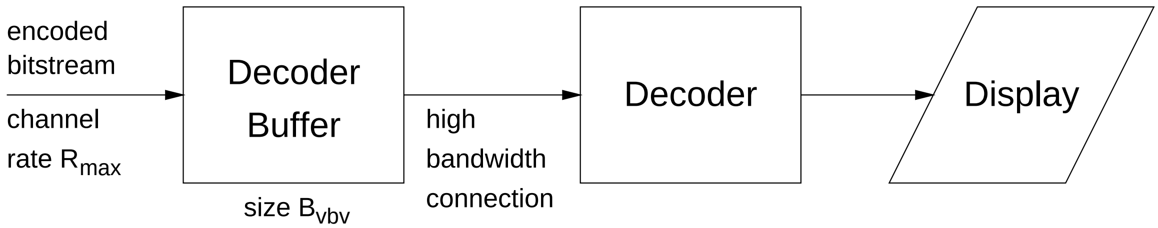


Figure 1: Block diagram of the MPEG Video Buffering Verifier.

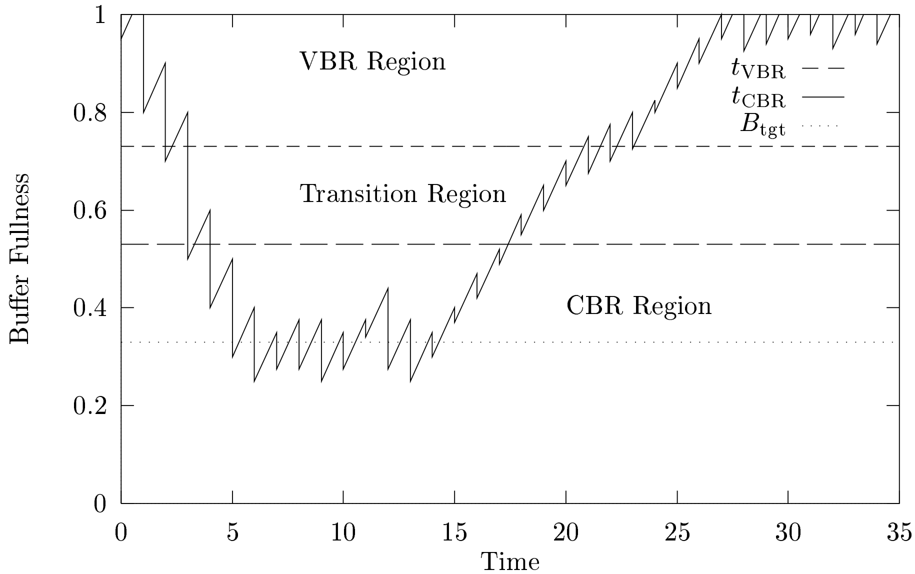
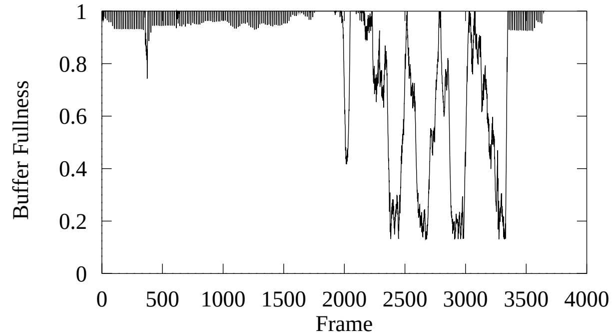


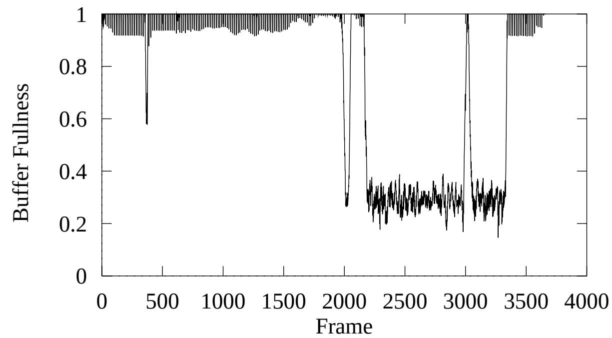
Figure 2: Illustration of the single-pass VBR algorithm. The algorithm initially operates in VBR mode, switches to CBR mode when $B_{\text{f}} \leq t_{\text{CBR}}$ and switches back to VBR mode when $B_{\text{f}} > t_{\text{VBR}}$.

Table 1: Results of encoding simulations.

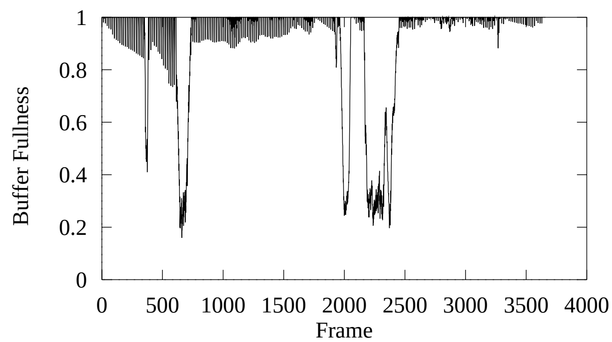
Rate Control Algorithm	Avg. PSNR (dB)	Std. dev. of PSNR	Avg. Nom. Q	Std. dev. of Nom. Q	Max. Nom. Q	Min. Nom. Q
Lexicographic VBR	33.09	2.54	11.74	1.97	16.84	9.02
Single-Pass VBR	32.76	3.09	12.87	4.20	33.59	10.50
Controlled VBR	32.93	4.70	14.59	8.40	29.64	2.14
TM5 CBR	33.28	5.79	16.10	12.81	53.71	1.96



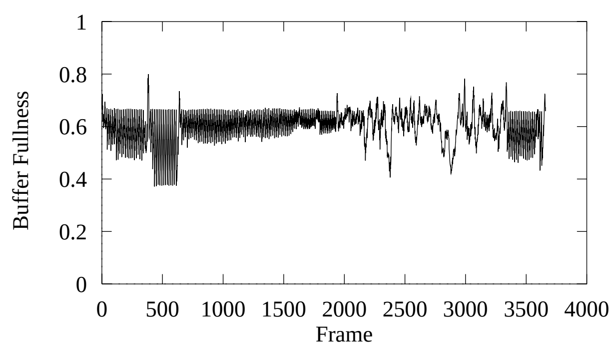
(a) Lexicographic VBR



(b) Single-Pass VBR

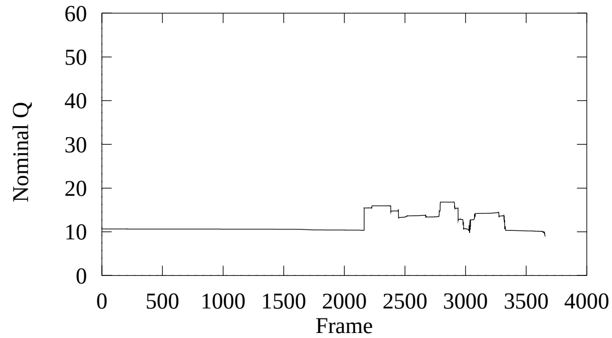


(c) Controlled VBR

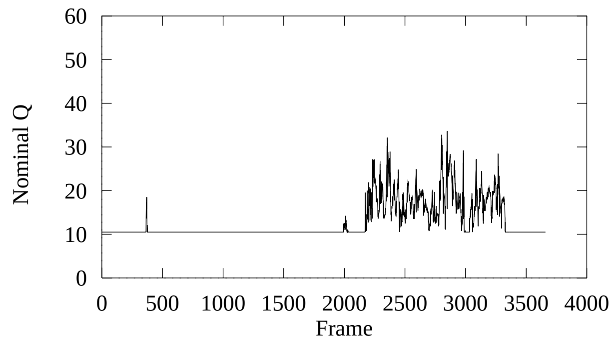


(d) TM5 CBR

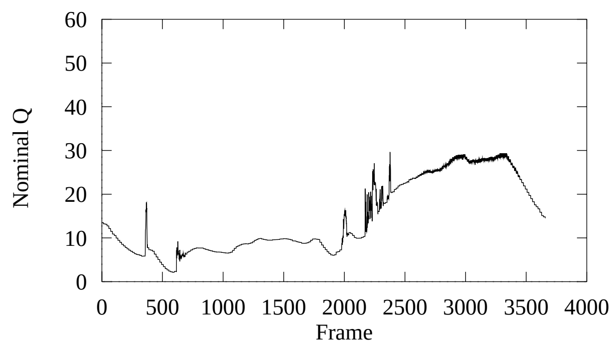
Figure 3: Evolution of normalized buffer fullness for a) lexicographically optimal VBR, b) single-pass VBR without long-term rate control, c) single-pass VBR with long-term rate control, and d) TM5 CBR.



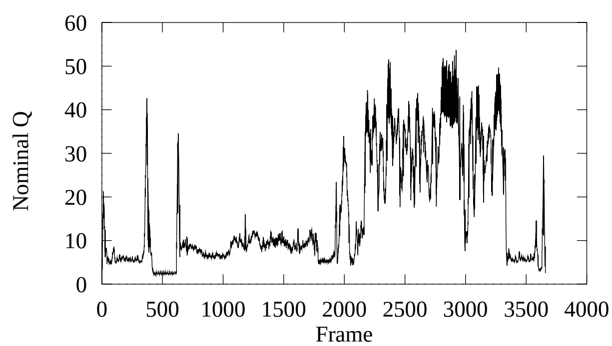
(a) Lexicographic VBR



(b) Single-Pass VBR

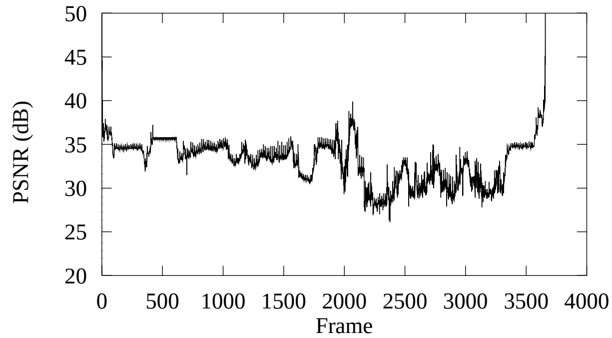


(c) Controlled VBR

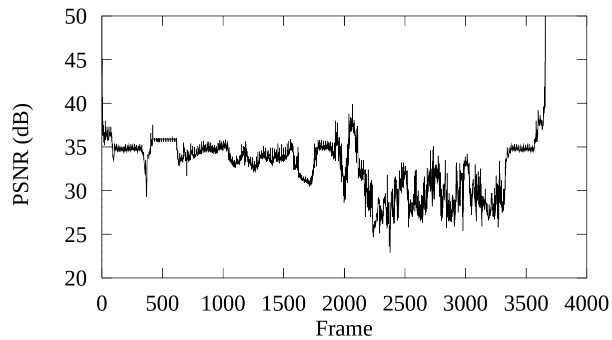


(d) TM5 CBR

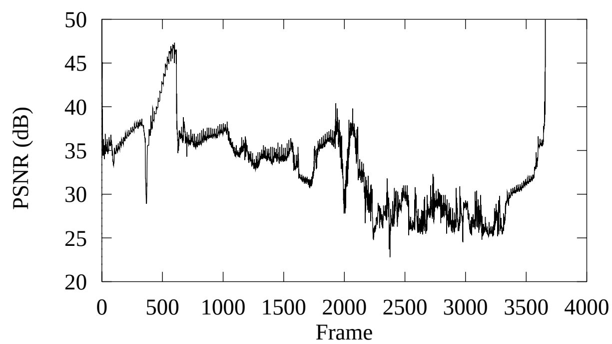
Figure 4: Nominal quantization for a) lexicographically optimal VBR, b) single-pass VBR without long-term rate control, c) single-pass VBR with long-term rate control, and d) TM5 CBR.



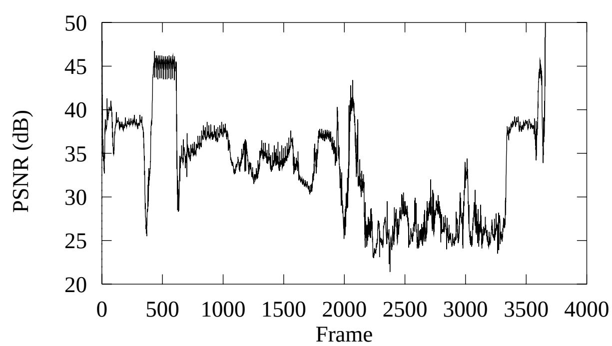
(a) Lexicographic VBR



(b) Single-Pass VBR



(c) Controlled VBR



(d) TM5 CBR

Figure 5: PSNR for a) lexicographically optimal VBR, b) single-pass VBR without long-term rate control, c) single-pass VBR with long-term rate control, and d) TM5 CBR.