Constant Quality Constrained Bit Allocation for Leaky Prediction Based FGS Video Streaming

Jianhua Wu[a], Jianfei Cai[a] and Chang Wen Chen[b]

[a] School of Computer Engineering,
Nanyang Technological University,
Block N4, Nanyang Avenue, Singapore 639798
[b]Department of Electrical and Computer Engineering,
Florida Institute of Technology,
Melbourne, FL 32901

ABSTRACT

Leaky prediction based FGS (Fine Granularity Scalability) can achieve better coding efficiency than the baseline FGS. However, for leaky prediction based FGS (L-FGS), constant quality constrained bit allocation, i.e., how to optimally allocate bits given the current channel bandwidth, is still an open problem. In this paper, based on the accurate R-D (Rate-Distortion) model developed in our previous work, we propose a constant quality constrained bit allocation scheme for L-FGS. The proposed scheme is a combination of offline and online processes. During the offline stage, we perform the L-FGS encoding and collect the necessary feature information. At the online stage, given the transmission bandwidth at that time, we quickly estimate the R-D curves of a sequence of consecutive video frames based on our previously developed R-D model and then perform the corresponding bit allocation using a sliding window technique. Experimental results show that our proposed bit allocation algorithm can achieve much more smooth video quality than the traditional uniform bit allocation under both CBR (constant bit rate) and VBR (variable bit rate) channels.

Keywords: Constant quality, bit allocation, rate-distortion, fine granularity scalability, leaky prediction

1. INTRODUCTION

With the development of multimedia communication techniques, video streaming applications over Internet and wireless networks become more and more popular in the recent years. For video streaming applications, heterogeneity is a very challenging problem. The heterogeneity may come from different users' requirements, different receiver devices, different access networks, different network conditions and etc.[1] Therefore, it is desired that the video coding rate can be adaptively adjusted according to the the particular condition. Scalable video coding is a common approach for providing rate adaptation.

Among various scalable video coding schemes,[2–4] FGS (Fine Granularity Scalability)[5] is the scalable coding technique adopted in MPEG-4.[6] Like other layered scalable video coding techniques, the basic idea of MPEG-4 FGS is to encode a video sequence into two layers: a base layer and an enhancement layer. The base layer is coded with the same method as non-scalable coding schemes such as MPEG-4, while the FGS enhancement layer is coded bitplane by bitplane. There is no temporal prediction in the FGS enhancement layer, which can prevent successive frames from error propagation at the cost of lower coding efficiency. Compared with traditional scalable coding schemes, the advantage of FGS video coding is that the enhancement layer can be truncated at any position, which provides fine granularity scalability.

To improve the coding efficiency of FGS, recently, many schemes have been proposed.[7–9] The common idea of these schemes is to make a better balance between the coding efficiency and error robustness by smartly introducing high quality reference frames to remove the temporal redundancy for the enhancement layer. Among all these schemes, the robust FGS (RFGS)[9] is the most representative technique, which employs two parameters,

Email: pg02320384@ntu.edu.sg (Jianhua Wu), asjfcai@ntu.edu.sg (Jianfei Cai), cchen@fit.edu (Chang Wen Chen).
This research is partially supported by Singapore A*STAR SERC Grant (032 101 0006).

Visual Communications and Image Processing 2005, edited by Shipeng Li,
Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proc. of SPIE Vol. 5960
(SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.633507

i.e., the number of bitplanes and the amount of predictive leak, to control the construction of reference frames for the tradeoff among the coding efficiency and the amount of drift. In this paper, we denote such leaky prediction[9–11] based FGS coding as L-FGS.

For video streaming applications, although rate adaptation can be easily realized in FGS video coding by bit truncation, there is a fundamental problem, i.e., how to truncate bits given the channel bandwidth. The simplest method is the uniform bit truncation.[5] However, this will cause low coding efficiency and video quality fluctuation among adjacent frames due to the non-stationary property of video signals, i.e., the flickering artifact. Therefore, it is desired to have optimal bit truncation/allocation among video frames in order to achieve constant video quality and improve the coding efficiency. Note that constant quality constrained bit allocation on the other hand will cause rate fluctuation which can be solved by carefully choosing the buffer size and the pre-loading time in video streaming applications.

In the past few years, many constant quality constrained bit allocation schemes[12, 13] have been proposed for the FGS video coding. In,[12] Zhang et al. constructed accurate R-D curves for the baseline FGS by extracting the R-D points during the encoding process followed by linear interpolation between neighbor R-D points. Based on the obtained R-D curves, a sliding window bit allocation scheme was proposed to realize the aim of constant quality. Zhao et al.[13] also proposed a constant quality rate control scheme for MPEG-4 FGS video. Compared with,[12] the work in[13] performed the constant quality rate control for both the base layer and the enhancement layer.

However, constant quality constrained bit allocation for L-FGS is still an open problem. The difficulty lies in obtaining accurate R-D curves for L-FGS due to the dependency among video frames. In our previous work,[14] we have proposed a novel rate-distortion (R-D) model for L-FGS video coding. This proposed R-D model considers not only the distortion introduced in the current frame and the propagated distortion from the reference frame due to rate adaptation, but also the correlation between them. In this paper, based on our previously developed R-D model, we propose a constant quality constrained bit allocation scheme for L-FGS. The proposed scheme is a combination of offline and online processes. During the offline stage, we perform the L-FGS encoding and collect the necessary feature information for the later online R-D estimation. At the online stage, given the transmission bandwidth at that time, we quickly estimate the R-D curves of a sequence of consecutive video frames based on the proposed R-D model and then perform the corresponding bit allocation using the sliding window technique.

The rest of this paper is organized as follows. Section 2 gives an overview of the leaky prediction based FGS. Section 3 gives an brief introduction of our previously proposed R-D model for L-FGS. Based on the R-D model, Section 4 describes the proposed constant quality constrained bit allocation scheme for L-FGS and Section 5 shows the experimental results. Finally, Section 6 concludes this paper.

2. OVERVIEW OF LEAKY PREDICTION BASED FGS

Using symbol definitions in Table 1, in a typical L-FGS system, the base layer residue $e_B(n,i)$ is obtained by

$$e_B(n,i) = F(n,i) - \hat{F}_B(n-1,j), \tag{1}$$

where $\hat{F}_B(n-1,j)$ is the motion-compensation reference frame. $e_B(n,i)$ and motion vectors are compressed into the base layer. Correspondingly, the base layer reconstruction, $\hat{F}_B(n,i)$, is given by

$$\hat{F}_B(n,i) = \hat{F}_B(n-1,j) + \hat{e}_B(n,i). \tag{2}$$

After the prediction from the base layer, the enhancement layer data can be represented as

$$\begin{aligned} F_E(n,i) &= F(n,i) - \hat{F}_B(n,i) \\ &= F(n,i) - \hat{F}_B(n-1,j) - \hat{e}_B(n,i). \end{aligned} \tag{3}$$

In the baseline FGS,[5] the enhancement layer data is directly compressed into the enhancement layer, i.e,

$$e_E(n,i) = F_E(n,i). \tag{4}$$

Table 1. The symbol definitions.

$F(n,i)$	:	The original value of pixel i in the n-th video frame.
$\tilde{F}(n,i)$	:	The received high quality value of pixel i in the n-th video frame at the decoder.
$\hat{F}_B(n,i)$	:	The base layer reconstruction value of pixel i in the n-th video frame in the base layer prediction loop at the encoder.
$e_B(n,i)$	:	The base layer residue value of pixel i in the n-th video frame.
$\hat{e}_B(n,i)$	:	The reconstructed base layer residue.
$F_E(n,i)$	:	The enhancement layer data after the prediction from the base layer.
$\hat{F}_E(n,i)$	:	The reconstructed enhancement layer data in the enhancement layer prediction loop at the encoder.
$\tilde{F}_E(n,i)$	:	The reconstructed enhancement layer data at the decoder.
$\hat{F}^p_E(n,i)$	:	The partial data of the reconstructed enhancement layer data, which is used in the enhancement layer prediction loop at the encoder.
$\tilde{F}^p_E(n,i)$	:	the partial data of the reconstructed enhancement layer data, which will be used in the enhancement layer prediction loop at the decoder.
$e_E(n,i)$	:	The enhancement layer residue after all the predictions, including the predictions from both the base layer and the enhancement layer.
$\hat{e}_E(n,i)$	:	The reconstructed enhancement layer residue at the encoder.
$\tilde{e}_E(n,i)$	:	The enhancement layer residue received at the decoder.

However, in the L-FGS, the high quality reference frame is introduced in the enhancement layer with a leaky factor α_n, and only the partial data of the reconstructed enhancement layer frame is used as the reference frame to make a better tradeoff between the coding efficiency and the robustness to drift errors.[9] In L-FGS, the enhancement layer residue $e_E(n,i)$ can be expressed as

$$e_E(n,i) = F(n,i) - \hat{e}_B(n,i) - (\hat{F}_B(n-1,j) + \alpha_{n-1}\hat{F}^p_E(n-1,j)), \tag{5}$$

where the same motion vectors generated in the base layer encoding are applied to the enhancement layer prediction coding in order to avoid re-conducting the most time-consuming motion estimation process. Note that $\hat{F}^p_E(n,i)$ is the partial data of $\hat{F}_E(n,i)$. Combining with Eqn. (3) , we simplify $e_E(n,i)$ as

$$e_E(n,i) = F_E(n,i) - \alpha_{n-1}\hat{F}^p_E(n-1,j) \tag{6}$$

Correspondingly, the enhancement layer reconstruction at the encoder is

$$\hat{F}_E(n,i) = \alpha_{n-1}\hat{F}^p_E(n-1,j) + \hat{e}_E(n,i), \tag{7}$$

which will be stored in the buffer for the encoding of the next enhancement layer frame.

Similar as in,[14] in this paper, we only consider introducing high quality reference frames in the enhancement layer, and keep the same encoding structure in the base layer as the baseline FGS. In addition, we assume that the bandwidth is always enough for transmitting the entire base layer and thus the truncation for rate adaptation only happens in the enhancement layer. H.263+ instead of MPEG-4 is employed to encode the base layer for simplicity. We only consider encoding video sequences with a pattern of one I-frame followed by all P-frames and TMN8 is used as the rate control scheme for the base layer. The L-FGS enhancement layer is encoded bitplane by bitplane, the same as that in the MPEG-4 FGS.

3. RATE-DISTORTION ANALYSIS OF L-FGS

During the transmission, the L-FGS enhancement layer bitstream will be truncated and only partial enhancement layer data will be delivered to the receiver to adapt to the channel bandwidth. In order to perform the optimal bit allocation in the enhancement layer, the key problem is to have the accurate R-D function for the L-FGS video coding. This section will give a brief introduction of our previously developed R-D model for the L-FGS video coding.[14]

Let $D(n)$ denote the overall distortion, based on the assumption that the whole base layer of L-FGS can be completely received and decoded, the distortion $D(n)$ comes from two parts, i.e., the distortion produced by truncating bits in the current enhancement layer frame, denoted as $D_I(n)$, and the distortion propagated from the previous frame due to the leaky prediction, denoted as $D_P(n-1)$. $D_I(n)$ and $D_P(n-1)$ are defined as

$$D_I(n) = E\{[F_E(n,i) - \alpha_{n-1}\hat{F}_E^p(n-1,j) - \tilde{e}_E(n,i)]^2\}, \tag{8}$$

$$D_P(n-1) = E\{[\hat{F}_E^p(n-1,i) - \tilde{F}_E^p(n-1,i)]^2\}, \tag{9}$$

where $E\{\}$ represents the average values over all pixels and $\hat{F}_E^p(n-1,j)$ is the motion-compensation reference frame of $F_E(n,i)$.

In,[14] it is found that $D(n)$ can be approximated as

$$D(n) = D_I(n) + \alpha_{n-1}^2\rho_{n-1}D_P(n-1) + (a_n + b_n\sqrt{D_I(n)})\sqrt{D_P(n-1)}, \tag{10}$$

where a_n and b_n are the constants, and a_n and b_n are different for different frames. ρ_{n-1} is a constant describing the motion randomness of the video scene, which can be calculated as

$$\rho_{n-1} = \frac{E\{[\hat{F}_E^p(n-1,j) - \tilde{F}_E^p(n-1,j)]^2\}}{D_P(n-1)}. \tag{11}$$

$D_I(n)$ can be calculated using the similar method employed in,[12,15] i.e., the linear interpolation technique, which can be performed at the offline encoding stage. In order to compute $D_P(n-1)$, two situations should be considered.

(1) The amount of the allocated bits in the enhancement layer is smaller than that for the partial leaky prediction, i.e., $\tilde{F}_E(n-1,i) = \tilde{F}_E^p(n-1,i)$. In this case, $D_P(n-1)$ can be calculated as

$$D_P(n-1) = D(n-1) - E\{[F_E(n-1,i) - \hat{F}_E^p(n-1,i)]^2\} - (c_{n-1} + d_{n-1}\sqrt{D(n-1)}), \tag{12}$$

where c_{n-1} and d_{n-1} are two constants, and the value of $E\{[F_E(n-1,i) - \hat{F}_E^p(n-1,i)]^2\}$ can be computed in the offline stage.

(2) The amount of the allocated bits in the enhancement layer is larger than or equal to that for the partial leaky prediction. Thus, $D_P(n-1)$ can be calculated as

$$\begin{aligned} D_P(n-1) &= E\{[\alpha_{n-2}\hat{F}_E^p(n-2,j) - \alpha_{n-2}\tilde{F}_E^p(n-2,j)]^2\} \\ &= \alpha_{n-2}^2\rho_{n-2}D_P(n-2). \end{aligned} \tag{13}$$

Note that, since the first frame (Frame 0) is intra-coded, $D(0) = D_I(0)$ and $D_P(0)$ can be calculated in the offline stage. At the online stage, based on the R-D model described above, we can estimate $D(n)$ for any value of n given the current allocated bandwidth.

4. CONSTANT QUALITY CONSTRAINED BIT ALLOCATION

In this section, we will apply the R-D model described in the previous section to solve the problem of constant quality constrained bit allocation for L-FGS.

Since the base layer is pre-encoded at the fixed bit rate, i.e., the lower bound of a bandwidth range, we only consider the bit allocation in the enhancement layer. We assume the transportation channels are piece-wise static channels, covering both the considered CBR and VBR channels. The constant quality constrained bit allocation problem can be formulated as

$$\min \sum_{j=i+1}^{i+W_i-1} ||D(j) - D(j-1)||, \quad \text{subject to} \sum_{j=i}^{i+W_i-1} B_j \le \frac{R_i \cdot W_i}{F}, \tag{14}$$

where i is the index of the current frame, F is the frame rate, W_i is the current sliding window size (in number of frames), and R_i is the current channel bit rate (in kbps). Note that at the time of allocating bits for the i-th frame, we only know the available channel bandwidth R_i at that time and we assume R_i remains unchanged over the entire sliding window W_i. This assumption is reasonable for piece-wise static channels, in which there exists strong correlation among adjacent time intervals.

Similar to,[13] we develop a sliding window scheme with bisection search to solve the problem shown in Eqn. (14). The basic idea is that, in each sliding window, we set a distortion D and make each frame have a distortion as close to D as possible by choosing appropriate rates. We repeat this procedure until a minimum distortion under the rate constraint is found. The detailed algorithm is described as follows.

Step 1: Initialize $i = 1$ and $\delta_0 = 0$, where δ_i is used to denote the difference between the total allocated number of bits and the total available channel bandwidth up to the frame i.

Step 2: Set $D_{low} = 0$, and assign the minimum distortion and the maximum distortion among all the base layer frames within the current sliding window W_i to D_{temp} and D_{high}, respectively.

Step 3: Calculate the target number of bits $B_j, j = i, \cdots, i + W_i - 1$, for each frame in the sliding window to make D_j equal to or closest to D_{temp} based on the R-D model described in the previous section.

Step 4: If $\sum_{j=i}^{i+W_i-1} B_j < \frac{R_i \cdot W_i}{F} - \delta_{i-1}$, set $D_{high} = D_{temp}$ and $D_{temp} = \frac{D_{low}+D_{high}}{2}$. Or else, If $\sum_{j=i}^{i+W_i-1} B_j > \frac{R_i \cdot W_i}{F} - \delta_{i-1}$, set $D_{low} = D_{temp}$ and $D_{temp} = \frac{D_{low}+D_{high}}{2}$.

Step 5: If $|\sum_{j=i}^{i+W_i-1} B_j - \frac{R_i \cdot W_i}{F} + \delta_{i-1}| > Mindiff \cdot W_i$, where $Mindiff$ is the predefined threshold which is used to control the rate adaptation accuracy, return to Step 3. Otherwise, the current value of B_i is used as the allocated number of bits for the frame i.

Step 6: Move to the next window. Update $\delta_i = \delta_{i-1} + B_i - \frac{R_i}{F}$ and $i = i + 1$. If there is any frame left, go to Step 2; otherwise stop.

Note that in this algorithm, for simplicity, we only consider the optimal bit allocation among all P frames and just assign the available bandwidth to the I-frame.

5. EXPERIMENTAL RESULTS

In this section, we conduct experiments to test the performance of the proposed bit allocation scheme. The experiments are performed on two QCIF format video sequences. The first video sequence is the Foreman sequence with 300 frames, which contains large facial movements and camera panning at the end. The second one is the Akiyo sequence with 300 frames, which contains low activity (slow motion). Both video sequences are coded at 10 fps. The base layers are coded at 32 kbps for the Foreman sequence and 16 kbps for the Akiyo sequence, respectively. In the enhancement layer encoding of the Foreman sequence, for each frame, an amount of 16 kbits is used in the partial prediction with a leaky factor of 0.7. For the Akiyo sequence, an amount of 11 kbits is used in the partial prediction with a leaky factor of 0.8 for each frame. It should be noted that although in this paper we use the same amount of bits for partial prediction and the same leaky factor for each frame, it will be interesting to apply different bits and different leaky factors to different frames, which is out of the scope of this paper.

We compare three bit allocation approaches: the uniform bit allocation method, our proposed bit allocation algorithm with $W = 40$ and our proposed bit allocation algorithm with $W = 80$, where W is the sliding window size. Fig. 1 shows the distortion results of adapting the QCIF Foreman sequence to CBR channels of 64 kbps and 128 kbps. Fig. 2 shows the distortion results of adapting the QCIF Akiyo sequence to CBR channels of 48 kbps and 80 kbps. It can be seen that our proposed constant quality constrained bit allocation scheme achieves much more smooth video quality than the uniform bit allocation. In addition, we can see that the case with larger sliding window size has smaller distortion variation. This is because for lager sliding window size the frame qualities are averaged over a wider range. Note that, for the case of adapting Foreman sequence to 64 kbps, there is an inconsistent hole as shown in Fig. 1. That is caused by the CBR coding in the base layer and insufficient bits for the enhancement layer. Table 2 gives a summary of adapting different video sequences to different CBR channels. "Distortion STD" (σ_D) denotes the standard deviation of the frame distortions, which is calculated as

$$\sigma_D = 10log_{10}\sqrt{\frac{1}{L}\sum_{i=1}^{L}(D(i) - \frac{1}{L}\sum_{j=1}^{L}D(j))^2}, \quad (15)$$

where L is the total number of frames. From the table, we can see that the average distortion of our proposed bit allocation is slightly lower than that of uniform bit allocation. The reduction of the distortion standard deviation (STD) is outstanding with at least 3 dB reduction.

Figs. 3 and 4 show the distortion performance of adapting the QCIF Foreman and Akiyo sequences to the time varying channels, respectively. Again, our proposed bit allocation scheme outperforms the uniform bit allocation method. It can be observed from the figures that our proposed bit allocation scheme can adapt to the time varying channels while still able to locally keep the consistent quality.

6. CONCLUSIONS

In this paper, we have proposed the constant quality constrained bit allocation scheme for L-FGS. The experimental results have demonstrated that the proposed scheme significantly outperforms the traditional uniform bit allocation and is able to achieve very consistent video quality. In the future, we will consider more practical issues such as the bit allocation under the constraints of pre-loading time and buffer size. We will also look into how to adaptively choose the leaky prediction factor.

REFERENCES

1. A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine* , pp. 18–29, March 2003.
2. T. K. Tan, K. K. Pang, and K. N. Ngan, "A frequency scalable coding scheme employing pyramid and subband techniques," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 203–207, April 1994.
3. H. Sun, W. Kwok, and J. W. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 191–199, April 1996.
4. M. Domanski, A. Luczak, and S. Mackowiak, "Spatial-temporal scalability for MPEG video coding," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 1088–1093, Oct. 2000.
5. W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 301–317, March 2001.
6. ISO/IEC JTC 1/SC 29/WG 11, "Information technology - generic coding of audio-visual objects - part 2: visual," *WG11 N2688* , March 1999.
7. F. Wu, S. Li, and Y. Q. Zhang, "A framework of efficient progressive fine granularity scalable video coding," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 332–344, March 2001.
8. M. Schaar and H. Radha, "Motion-compensation based find-granular scalability (MC-FGS)," in *ISO/IEC JTC1/SC29/WG11, MPEG00/M6475*, Oct. 2000.
9. H. Huang, C. Wang, and T. Chiang, "A robust fine granularity scalability using trellis-based predictive leak," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 372–385, June 2002.
10. Y. Liu, Z. Li, P. Salama, and E. Delp, "A discussion of leaky prediction based scalable coding," in *IEEE ICME 2003*, pp. 565 – 568, Jul. 2003.
11. S. Han and B. Girod, "Robust and efficient scalable video coding with leaky prediction," in *IEEE ICIP 2002*, pp. II–41 – II–44, Sept. 2002.
12. X. M. Zhang, A. Vetro, Y. Q. Shi, and H. Sun, "Constant quality constrained rate allocation for FGS-coded video," *IEEE Trans. on Circuits and Systems for Video Technology* , pp. 121–130, Feb. 2003.
13. L. Zhao, J. W. Kim, and C.-C. Kuo, "Constant quality rate control for streaming MPEG-4 FGS video," in *IEEE ISCAS 2002*, pp. IV–544 – IV–547, May 2002.
14. J. Wu and J. Cai, "A novel rate-distortion model for leaky prediction based FGS video coding," in *IEEE PCM 2004*, pp. 673–680, Nov. 2004.
15. J. Wu and J. Cai, "Wireless FGS video transmission using adaptive mode selection and unequal error protection," in *SPIE VCIP 2004*, pp. 1305–1316, Jan. 2004.

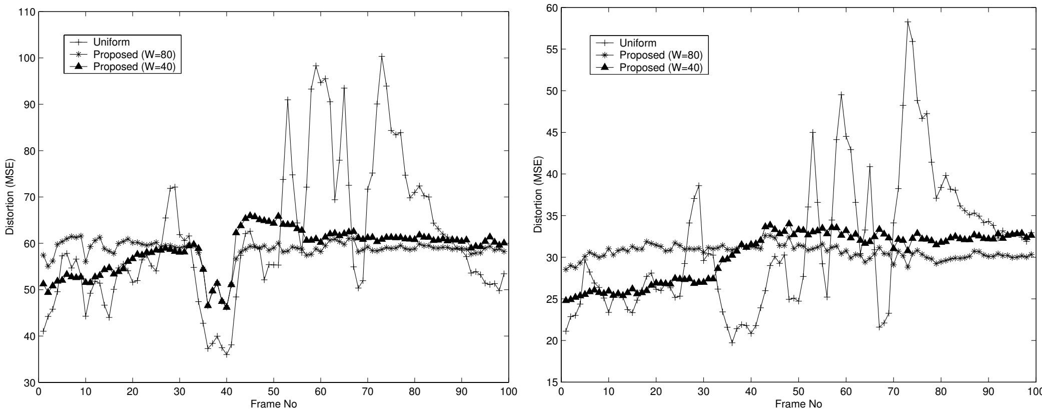

Figure 1. Distortion for adapting the L-FGS coded QCIF Foreman sequence to CBR channels. Left: QCIF Foreman at 64 kbps. Right: QCIF Foreman at 128 kbps.

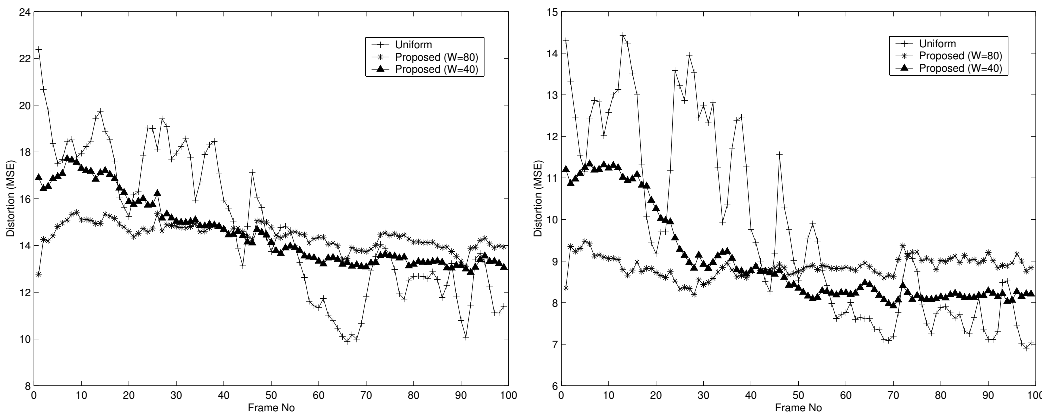

Figure 2. Distortion for adapting the L-FGS coded QCIF Akiyo sequence to CBR channels. Left: QCIF Akiyo at 48 kbps. Right: QCIF Akiyo at 80 kbps.

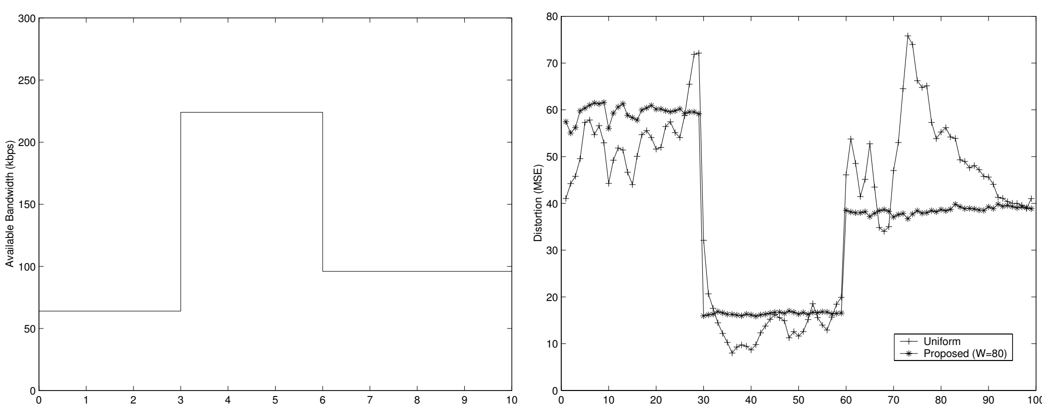

Figure 3. The simulation results for adapting the L-FGS coded Foreman sequence to the VBR channel. Left: Available bandwidth. Right: Distortion performance with window size=80.

Table 2. The performance comparison between the uniform bit allocation and our proposed scheme.

Video Sequence	Channel Rate (kbps)	Bit Allocation Scheme	Average Distortion (dB)	Distortion STD (dB)
Foreman	32+32	Uniform	17.88	11.80
		Proposed ($W = 40$)	17.68	6.64
		Proposed ($W = 80$)	17.67	4.47
	32+64	Uniform	16.40	10.16
		Proposed ($W = 40$)	16.31	5.03
		Proposed ($W = 80$)	16.31	-1.30
	32+96	Uniform	14.98	9.08
		Proposed ($W = 40$)	14.82	4.73
		Proposed ($W = 80$)	14.86	-1.00
Akiyo	16+32	Uniform	11.72	4.84
		Proposed ($W = 40$)	11.64	1.60
		Proposed ($W = 80$)	11.59	-2.92
	16+64	Uniform	9.88	3.56
		Proposed ($W = 40$)	9.54	0.46
		Proposed ($W = 80$)	9.48	-6.18
	16+96	Uniform	7.76	-1.59
		Proposed ($W = 40$)	7.65	-6.24
		Proposed ($W = 80$)	7.66	-7.41

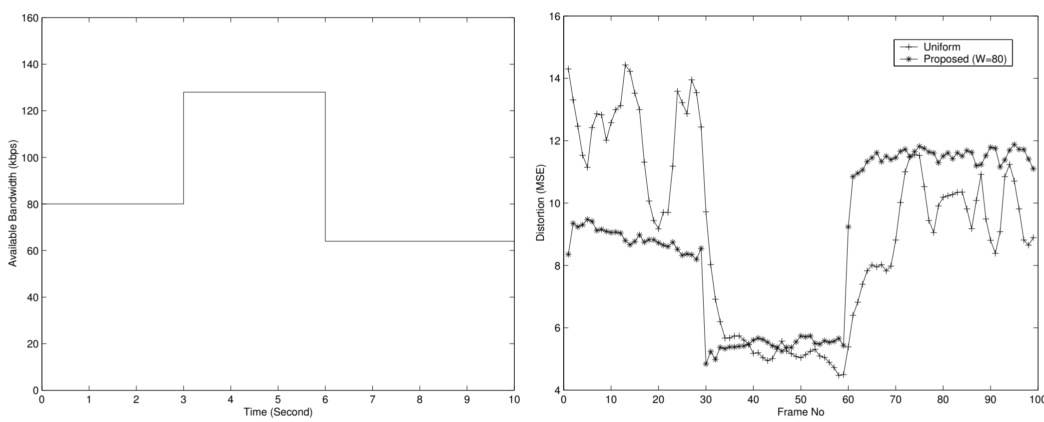

Figure 4. The simulation results for adapting the L-FGS coded Akiyo sequence to the VBR channel. Left: Available bandwidth. Right: Distortion performance with window size=80.