Lowpass Filtering of Rate-Distortion Functions for Quality Smoothing in Real-Time Video Communication

Zhihai He, *Member, IEEE,* Wenjun Zeng, *Senior Member, IEEE,* Chang Wen Chen, *Senior Member, IEEE*

***Abstract*— In variable-bit-rate (VBR) video coding, the video is pre-processed to collect sequence-level statistics, which are used for global bit allocation in the actual encoding stage to obtain a smoothed video presentation quality. However, in real-time video recording and network streaming, this type of two-pass encoding scheme is not allowed because the access to future frames and global statistics is not available. To address this issue, we introduce the concept of low-pass filtering of rate-distortion (R-D) functions and develop a smoothed rate control (SRC) framework for real-time video recording and streaming. Theoretically, we prove that, using a geometric averaging filter, the SRC algorithm is able to maintain a smoothed video presentation quality while achieving the target bit rate automatically. We also analyze the buffer requirement of the SRC algorithm in real-time video streaming, and propose a scheme to seamlessly integrate robust buffer control into the SRC framework. The proposed SRC algorithm has very low computational complexity and implementation cost. Our extensive experimental results demonstrate that the SRC algorithm significantly reduces the picture quality variation in the encoded video clips.**

***Index Terms*— Quality smoothing, variable bit rate, real-time video coding, bit allocation.**

I. Introduction

In digital video recording and compression, the encoding bit rate needs to be controlled so that the video storage size or transmission bandwidth constraint is satisfied. For a given bit budget, the ultimate goal of the rate control algorithm is to optimize the video presentation quality. To achieve a visually pleasing video presentation, not only does each video frame need to be encoded at the highest quality level, but also the frame-to-frame perceptual quality changes need to be smooth enough so that temporal artifacts, such as quality flicker and motion jerkiness, are minimized. In [1], video quality smoothing is formulated as a Lagrange minimization problem, where the quality smoothness is measured by the frame-to-frame variation of picture quality. To optimize the video quality under the bit rate constraint, a two-pass encoding scheme is often used by rate control algorithms. More specifically, in the first pass of pre-processing, the encoder collects coding complexity information of the entire video clip. During the second pass of actual encoding, based on this complexity information, the encoder performs global bit allocation to determine the encoding parameters, such as quantization parameter (QP) of each frame, so as to optimize the overall picture quality [1], [2], [3].

Such type of two-pass encoding scheme is not applicable to real-time video recording and streaming applications, including Personal Video Recording (PVR), digital camcorder, live video streaming and video conferencing, because the access to future frames and global statistics is not available. In other words, real-time video applications require one-pass video encoding. Within the one-pass encoding framework, without access to the coding characteristics of future frames, it is very difficult to maintain a smoothed video presentation quality while meeting the target encoding bit rate, because the encoder has no idea about how complicated the future scenes might be [4]. In low-delay constant-bit-rate (CBR) video coding, it is easy for the encoder to match the network bandwidth or total storage space by simply setting the bit rate target of each frame to be the instantaneous network bandwidth or the average storage space per frame [14]. However, the picture quality varies significantly from frame to frame due to the changes in scene activity. In this case, there is no control in temporal quality fluctuation at all. To optimize video presentation quality, especially the temporal quality smoothness, one typical approach is to perform frame-level bit allocation, as in the standard TM5 rate control [5]. The encoder sets the bit rate target for each GOP (group of pictures) to be the average available number of bits per GOP and allocates the bits among the frames within the GOP. TM5 assumes that the future frames have similar statistics as their previous frames and uses the previous statistics for bit allocation to optimize the picture quality, including temporal quality smoothness [5]. This type of approach has several disadvantages. First, the assumption of similar statistics itself may not hold, especially for videos with high motion and frequent scene changes. The bit allocation based on this assumption will lead to significant quality fluctuation within the GOP. Second, there is no mechanism available to guarantee the smoothness of GOP-to-GOP quality change, because each GOP carries out the bit allocation and quality optimization independently [5]. In [4], a different approach is proposed that uses a square-root formula to predict the coding bit rate from mean absolute difference (MAD). The ratio between the MAD value of the current frame and the average MAD value of all its previous encoded frames is then used to determine the target bit rate

Manuscript received September 30th, 2003; revised March 22nd, 2004.

Z. He is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211, USA (e-mail: HeZhi@missouri.edu).

W. Zeng is with the Department of Computer Sciences, University of Missouri, Columbia, MO 65211, USA (e-mail: zengw@missouri.edu).

C. W. Chen is with the Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne, FL 32901, USA (e-mail: cchen@fit.edu).

for the current frame. Although the scheme reduces the picture quality fluctuation, it does not provide an explicit and analytic way to maintain a smooth video presentation quality while meeting the overall bit rate target.

In this work, we introduce the concept of low-pass filtering of rate-distortion (R-D) functions and develop a smoothed rate control (SRC) framework for real-time video recording. Theoretically, we prove that, using a geometric averaging filter, the SRC algorithm is able to maintain a smoothed video presentation quality while achieving the target bit rate automatically. We also analyze the buffer size requirement of the SRC algorithm in real-time video streaming, and propose a scheme to seamlessly integrate robust buffer control into the SRC framework. We then conduct extensive simulations over various TV programs and movie clips to demonstrate the efficiency of the SRC algorithm. The proposed SRC algorithm has very low computational complexity and implementation cost.

The rest of this paper is organized as follows. In Section II, we explain the basic ideas of low-pass filtering of R-D functions and SRC. In Section III, we prove theoretically that SRC is able to achieve the target bit rate using a geometric averaging filter. Section IV explains how to construct the CBR R-D functions which are used for SRC. In Section V, we analyze the buffer requirement of the SRC algorithm in real-time video streaming and develop a buffer-constrained SRC algorithm. Section VI summarizes the major steps in the SRC algorithm and analyzes the computational complexity and implementation cost. Experimental results are presented in Section VII and some concluding remarks are given in Section VIII.

II. Lowpass Filtering of R-D Functions

In CBR video coding, the picture quality varies significantly from frame to frame, especially for videos with active scenes. Fig. 1-(A) shows the distortion of each frame, denoted by $D_C(n)$, of the "NBA" CIF (352× 288) video coded at 1.8 M bits per second (bps). Here, the frame target is set to be the average encoding bit rate, denoted by R_T. In other words, $R_C(n) = R_T$. It can be seen that there is a very large frame-to-frame quality fluctuation.

From the human visual system (HVS) point of view, smoothed video quality yields visually pleasing human perception, while quality flicker and temporal noise are very annoying in video presentation [17]. The basic idea of the proposed quality smoothing algorithm is to design a rate control scheme such that the output video quality changes very smoothly from frame to frame. We refer to this type of rate control algorithm as smoothed rate control (SRC). In signal processing, a common approach to obtain smoothness is to use lowpass filtering. In this work, we indeed apply the lowpass filtering approach to design an efficient SRC algorithm. Fig. 1-(B) (in solid line) shows the lowpass filtering output of the CBR distortion profile $\{D_C(n)\}$ of Fig. 1-(A). It can be seen that the output distortion profile, denoted by $\{D_S(n)\}$, is quite smooth. Let the corresponding encoding bit rate of each frame be $R_S(n)$. Obviously, $R_S(n)$ is not constant any more, as shown in Fig. 1-(C), and depends on the design of the lowpass filter. In video recording application, it is required that the average encoding bit rate matches exactly the available storage space or transmission bandwidth. This brings up an interesting issue: how to design a lowpass filter such that the run-time average of $R_S(n)$ is equal to R_T? This question will be answered in Section III. Furthermore, the SRC algorithm also needs to consider buffer compliance for real-time video streaming, which is to be discussed in Section V.

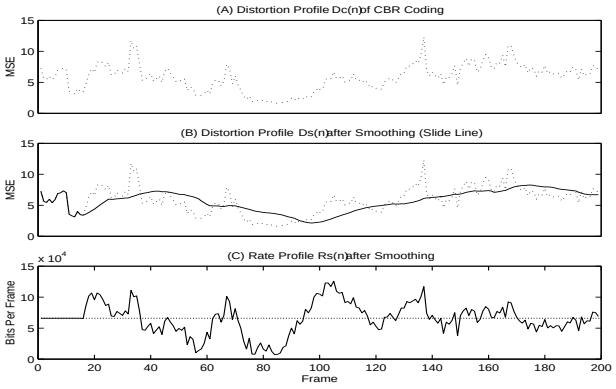

Fig. 1. Illustration of the basic idea for quality smoothing in the SRC algorithm.

III. Theoretical Analysis

In our SRC design, we apply the following lowpass filter to smooth out the CBR distortion profile $\{D_C(n)\}$,

$$\begin{aligned} D_S(n) &= \mathcal{L}[D_C(n)] \\ &= \prod_{i=1}^{M} [D_C(n-i)]^{a_i} \end{aligned} \quad (1)$$

where M is the filter length, $\sum_{i=1}^{M} a_i = 1$, and $a_i > 0$. Here, $\{a_i\}$ are weighting factors, controlling the relative importance of previous frames to the quality of the current frame. By default, we can set $a_i = \frac{1}{M}$. It can be seen that $\mathcal{L}[\cdot]$ is basically a non-linear geometric averaging filter. From the theoretical analysis in the following, we will see that using this lowpass filter, the target bit rate can be achieved automatically.

In our SRC scheme, the distortion level of the current frame is set to be the geometric average of the CBR distortion values of previous M frames. Let $R_S(n)$ be the encoding bit rate of frame n in the SRC mode, whose run-time average is defined as

$$\bar{R}_S[N] = \frac{1}{N} \sum_{n=1}^{N} R_S(n), \quad (2)$$

where N is the total number of encoded video frames. We need to show that when the encoded video clip is sufficiently long, i.e., when N is sufficiently large, the asymptotic value of $\bar{R}_S[N]$ approaches the target encoding bit rate R_T. In other words,

$$\lim_{N \to +\infty} \bar{R}_S[N] = R_T. \quad (3)$$

From Shannon's source coding theorem [10], [11], the R-D distortion function of a Gaussian source is given by

$$R(D) = \frac{1}{2}\log_2\frac{\sigma^2}{D}, \quad \text{or} \quad D(R) = \sigma^2 2^{-2R}, \tag{4}$$

where σ^2 is the picture variance. Due to scene activity fluctuations, σ^2 changes from frame to frame. Let $\sigma^2(n)$ be the variance of frame n, where $1 \le n \le N$. In CBR video coding, the coding bit rate $R_C(n)$ is set to be R_T. Therefore,

$$D_C(n) = \sigma^2(n)2^{-2R_T}. \tag{5}$$

From (1), we have

$$\begin{aligned} D_S(n) &= \prod_{i=1}^{M}[D_C(n-i)]^{a_i} &(6)\\ &= \prod_{i=1}^{M}[\sigma^2(n-i)2^{-2R_T}]^{a_i}. &(7)\end{aligned}$$

According to (4), the corresponding coding bit rate is given by

$$R_S(n) = \frac{1}{2}\log_2\frac{\sigma^2(n)}{D_S(n)}. \tag{8}$$

Thus, we have

$$\begin{aligned} &\bar{R}_S[N]\\ =\ & \frac{1}{N}\sum_{n=1}^{N} R_S(n)\\ =\ & \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log_2\frac{\sigma^2(n)}{D_S(n)}\\ =\ & \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log_2\frac{\sigma^2(n)}{\prod_{i=1}^{M}[\sigma^2(n-i)2^{-2R_T}]^{a_i}} &(9)\\ =\ & R_T + \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log_2\frac{\sigma^2(n)}{\prod_{i=1}^{M}[\sigma^2(n-i)]^{a_i}}\\ =\ & R_T + \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left[\log_2\sigma^2(n) - \sum_{i=1}^{M} a_i\log_2\sigma^2(n-i)\right]. \end{aligned}$$

Note that $\sigma^2(n)$ is bounded and

$$\lim_{N\to+\infty}\frac{1}{N}\sum_{n=1}^{N}[\log_2\sigma^2(n) - \log_2\sigma^2(n-i)] = 0, \tag{10}$$

for $0 \le i \le M$. Therefore, we have

$$\lim_{N\to+\infty}\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left[\log_2\sigma^2(n) - \sum_{i=1}^{M} a_i\log_2\sigma^2(n-i)\right] = 0,$$

and

$$\lim_{N\to+\infty}\bar{R}_S[N] = R_T. \tag{11}$$

This result tells us that if we use the geometric averaging filter $\mathcal{L}[\cdot]$ to smooth out the CBR distortion profile and determine the distortion level of the current video frame, the average encoding bit rate matches the target bit rate.

Notice that the R-D function in (4) and the above mathematical derivation is based on the Gaussian distribution. If we assume the input has a Laplacian distribution

$$p(x) = \frac{\lambda}{2}e^{-\lambda|x|}, \tag{12}$$

and the corresponding R-D function becomes,

$$R(D) = \log_2\frac{1}{\lambda D} = \log_2\frac{\sigma^2}{2D}. \tag{13}$$

where $\lambda = \frac{2}{\sigma^2}$, and σ^2 is the input variance [11]. If we replace (4) with (13) and follow the same procedure, we can show that the geometric averaging filter also automatically achieves the bit rate target for Laplacian sources.

Another interesting point to note is that if we use the following FIR lowpass filter

$$L(z) = \sum_{i=1}^{M} a_i z^{-i}, \tag{14}$$

which is the arithmetic average of previous M samples, following similar mathematical procedures, we can get

$$\bar{R}_S[N] = R_T + \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\left[\log_2\sigma^2(n) - \log_2\sum_{i=1}^{M} a_i\sigma^2(n-i)\right]. \tag{15}$$

Clearly, the arithmetic average FIR filter is conceptually simpler than the geometric one. Let $\delta[N]$ be the difference of $\bar{R}_S[N]$ between (9) and (15). We have

$$\delta[N] = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}H(n), \tag{16}$$

where

$$H(n) = \log_2\sum_{i=1}^{M} a_i\sigma^2(n-i) - \sum_{i=1}^{M} a_i\log_2\sigma^2(n-i). \tag{17}$$

From Jensen's inequality, we know $H(n) \ge 0$, which implies that the average encoding bit rate with the FIR filter $L(z)$ is always lower than the target R_T. However, our experimental results show that the difference $H(n)$ is often very small. Fig. 2-(A) plots the picture variance of a typical sports video clip with very high motion coded at 30 fps with a GOP size of 60. In Fig. 2-(B) we plot the relative difference in percentage $\mathcal{E}(n)$ defined as follows:

$$\mathcal{E}(n) = \frac{H(n)}{\log_2\sum_{i=1}^{M} a_i\sigma^2(n-i)} \times 100\% \tag{18}$$

It can be seen that the relative difference is very small, mostly less than 3%. This suggests that, in practice, we may also use the FIR arithmetic averaging filter for distortion smoothing, which is conceptually simpler and has lower implementation cost.

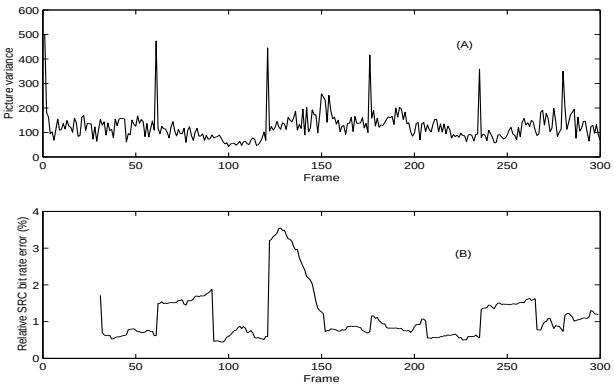

Fig. 2. Analysis of SRC bit rate error: (A) picture variance of each video frame; (B) plot of $\mathcal{E}(n)$.

IV. Construction of the CBR R-D Profile

In our SRC scheme, the video is encoded in a VBR fashion with a smoothed picture quality profile. More specifically, the video encoding is following the VBR R-D profile indicated by the solid lines in Fig. 1. Note that the video coding is one-pass. After a frame is encoded, what we can get is an R-D point $[R_S(n), D_S(n)]$ on the VBR profile, indicating the actual encoding bits and picture distortion. From (1), we can see that, in order to implement our idea of SRC, we need the CBR R-D profile. This implies that we need to construct the CBR R-D profile from the VBR one $[R_S(n), D_S(n)]$. To achieve such a goal, we introduce a linear rate model and explain how the CBR picture distortion can be estimated with this model.

A. Linear Rate Model

We use the simple and accurate linear rate model developed in our previous work [8], [15] to estimate the CBR R-D points. For the integrity of this paper, here, we give a brief review of the linear rate model. In conventional R-D studies [13], [14], the encoding bit rate and distortion, denoted by R and D, is considered as functions of the quantization parameter q, i.e., $R(q)$ and $D(q)$, respectively. In [15], it has been observed that the percentage of zeros among the quantized transform coefficients, denoted by ρ, plays a very important role in transform coding. Note that ρ monotonically increases with q, which implies that there is a one-to-one mapping between them. Therefore, mathematically, R and D are also functions of ρ, denoted by $R(\rho)$ and $D(\rho)$, respectively. It has been demonstrated both theoretically and experimentally that, in standard video coding systems, such as MPEG-2 [16], H.263 [6], and MPEG-4 [7], there is a linear relationship between the actual coding bit rate R and ρ, i.e.,

$$R(\rho) = \theta \cdot (1 - \rho), \tag{19}$$

where θ is a frame constant. The one-to-one mapping between q and ρ can be computed from the distribution of the DCT coefficients. Let us take the H.263 quantization for an example. Suppose $\mathcal{D}_0(x)$ and $\mathcal{D}_1(x)$ are the distributions of the DCT coefficients in the intra and inter macroblocks (MBs). For a given quantization parameter q, the corresponding percentage of zeros ρ can be computed as follows,

$$\rho(q) = \frac{1}{K} \sum_{|x| \le q} \mathcal{D}_0(x) + \frac{1}{K} \sum_{|x| \le 1.25q} \mathcal{D}_1(x), \tag{20}$$

where K is the number of coefficients in the current video frame. In SRC, when frame n is encoded, we know the actual encoding bit rate $R_S(n)$ and the percentage of zeros produced by the encoder, $\rho_S(n)$. According to (19), to achieve the encoding bit rate of R_T in CBR coding, the encoder needs to generate the following percentage of zeros

$$\rho_C(n) = 1 - \frac{R_T}{R_S(n)}[1 - \rho_S(n)]. \tag{21}$$

Using the one-to-one mapping in (20), we can compute the quantization parameter q_C such that

$$\rho(q_C) = \rho_C(n). \tag{22}$$

In other words, if the quantization parameter q_C is used, the encoder should be able to achieve the CBR coding bit rate R_T.

B. Computing the CBR Distortion

Using the linear rate model and VBR coding R-D statistics, we can determine the encoder quantization parameter q_C that is able to achieve the target bit rate in the CBR coding mode. From q_C, we can compute the corresponding CBR distortion $D_C(n)$ Again, let us now take the H.263 quantization for an example. Let $S_0(n)$ and $S_1(n)$ be the sets of coefficients in Intra and Inter MB's in frame n, and $\mathcal{D}_0(n,x)$ and $\mathcal{D}_1(n,x)$ be their distributions, respectively. For a given quantization parameter q, the corresponding distortion is given by

$$\begin{aligned} D(n;q) &= \sum_{x \in S_0(n)} \mathcal{D}_0(n,x)[x - \mathcal{Q}_0(x,q)]^2 \\ &+ \sum_{x \in S_1(n)} \mathcal{D}_0(n,x)[x - \mathcal{Q}_1(x,q)]^2, \end{aligned} \tag{23}$$

where $\mathcal{Q}_0(x,q)$ and $\mathcal{Q}_1(x,q)$ are the reconstruction levels of x in Intra and Inter quantization modes, respectively [6]. The CBR distortion $D_C(n)$ is then given by

$$D_C(n) = D(n, q_C). \tag{24}$$

With these estimated rate and distortion parameters, we can construct the CBR R-D profile. Note that there is no approximation in the distortion model in (23), therefore, the reconstruction error of the CBR R-D profile comes only from the linear rate model. According to the extensive simulation results in [15], the linear rate model in (19) is accurate, with a relative estimation error less than 5%. In this work, according to our simulations on various video sequences, the relative reconstruction error of the CBR R-D profile is about 5-8%.

V. SRC with Buffer Constraints for Video Streaming

A. Buffer Constraints in VBR Video Streaming

As we have indicated, using the linear rate model and the distortion formula developed in Section IV, we can construct

the CBR R-D functions and perform the lowpass smoothing of the distortion profile as discussed in Section II. Note that there is no constraint on the rate profile $\{R_S(n)\}$ during the distortion smoothing. In other words, $\{R_S(n)\}$ could have arbitrarily large fluctuation when the scene activity changes significantly. In video recording for offline local playback, such as movie compression, Personal Video Recording (PVR), and TV program replay, the compressed video data is saved in a local storage. Large frame-to-frame rate fluctuation for the purpose of quality smoothness is allowed. The only constraint on $\{R_S(n)\}$ is that sum of $\{R_S(n)\}$ is equal to the available storage space, which has been already guaranteed by our theoretical analysis in Section III.

In real-time video streaming, each input frame is compressed by the video encoder. The compressed bit stream flows into an encoder buffer which is drained by the network channel. After traveling through the network transmission channel, the bit stream arrives at a decoder buffer. The video decoder fetches the bit stream, decodes and displays the video frame. In such a real-time system, the video server (or encoder) and receiver operates in a synchronized fashion. The end-to-end delay, or the overall travel time of each video frame between the moment of its entry into the encoder and its display time at the receiver, needs to be a constant, denoted by Δ. In general, we have

$$\Delta = \Delta_{enc} + \Delta_{eb} + \Delta_t + \Delta_{dec} + \Delta_{db}, \tag{25}$$

where Δ_{enc} and Δ_{dec} are the frame encoding and decoding time; Δ_{eb} and Δ_{db} are the encoder and decoder buffer delays, and Δ_t represents the network transmission time [9]. In video streaming, especially one-way streaming, such as video-on-demand, the buffer delays are significantly larger than the frame encoding, decoding and network transmission time. Therefore, the end-to-end delay is mainly affected by the buffer delays, which are determined by the encoder and decoder buffer sizes, denoted by W_e and W_d, respectively. Let

$$L = \frac{\Delta_{eb} + \Delta_{db}}{\tau}, \tag{26}$$

where τ is the frame interval. Let $C(i)$ be the channel transmission rate at frame time i. The encoder and decoder buffer occupancies at frame time n, denoted by $B_e(n)$ and $B_d(n)$ are given by

$$B_e(n) = \sum_{i=1}^{n} R_S(i) - \sum_{i=1}^{n} C(i) \tag{27}$$

$$B_d(n) = \begin{cases} \sum_{i=1}^{n} C(i) - \sum_{i=1}^{n-L} R_S(i), & \text{when} \quad i \geq L \\ \sum_{i=1}^{n} C(i), & \text{when} \quad i < L. \end{cases} \tag{28}$$

When $B_e(n) > W_e$ or $B_d(n) > W_d$, the buffer overflows and the additional video data will be dropped. The dropped data will cause decoding failure or picture reconstruction error at the receiving end. When $B_e(n) \leq 0$ the encoder buffer underflows and the network channel is under utilized since there are no bits to transmit. When $B_d(n) < 0$, the decoder buffer underflows. The decoder has to pause the decoding process and waits for the bit stream to arrive, which may cause jerkiness in the video presentation. Therefore, in our SRC design, we need to avoid buffer overflow and underflow at both encoder and decoder sides. From (27) and (28), we observe that

$$\begin{aligned} B_d(n+L) &= \sum_{i=1}^{n+L} C(i) - \sum_{i=1}^{n} R_S(i) \\ &= \sum_{i=n+1}^{n+L} C(i) - [\sum_{i=1}^{n} R_S(i) - \sum_{i=1}^{n} C(i)] \\ &= \sum_{i=n+1}^{n+L} C(i) - B_e(n). \end{aligned} \tag{29}$$

This is the so-called mirror effect of buffer occupancy between the encoder and decoder. This implies that we only need to control the encoder buffer.

B. Maximum Buffer Size in SRC

In live video streaming, the available network transmission rate $C(i)$ is often varying due to other network traffic [9]. We can often set the encoder bit rate target R_T to be the average network transmission rate. In other words, we let $\sum_{i=1}^{n} C(i) = nR_T$. From (9) and (27), we have

$$\begin{aligned} B_e(n) &= \sum_{i=1}^{n} R_S(i) - \sum_{i=1}^{n} C(i) \\ &= \sum_{i=1}^{n} \frac{1}{2} \left[\log_2 \sigma^2(i) - \sum_{j=1}^{M} a_j \log_2 \sigma^2(i-j) \right] \\ &= \frac{1}{2} \sum_{j=1}^{M} a_j \left(\sum_{i=1}^{n} [\log_2 \sigma^2(i) - \log_2 \sigma^2(i-j)] \right) \\ &= \frac{1}{2} \sum_{j=1}^{M} a_j \left(\sum_{k=0}^{j-1} [\log_2 \sigma^2(n-k) - \log_2 \sigma^2(1)] \right). \end{aligned} \tag{30}$$

Here we assume $\log_2 \sigma^2(i) = \log_2 \sigma^2(1)$ for $i < 1$. Note that $\{\sigma^2(i)\}$ is bounded, and there exists a constant Θ such that $|\log_2 \sigma^2(i)| \leq \Theta$. Therefore,

$$|B_e(n)| \leq \frac{1}{2} \Theta \sum_{j=1}^{M} a_j \cdot j. \tag{31}$$

Let

$$W_{max} = \Theta \sum_{j=1}^{M} a_j \cdot j. \tag{32}$$

It can be seen that W_{max} is the maximum encoder buffer size that is needed for video streaming with SRC, because we can let the encoder buffer accumulate bits to the buffer level of $\frac{W_{max}}{2}$ during the initial buffering stage and then start the network transmission. From (31), we know $|B_e(n)| \leq \frac{W_{max}}{2}$. Therefore, the encoder will never experience buffer overflow and underflow. If an equally weighted geometric averaging filter is used in distortion smoothing, in other words, $a_i = \frac{1}{M}$, we have

$$W_{max} = \frac{M+1}{2} \Theta, \tag{33}$$

which suggests that larger smoothing window (M) and larger variation in scene activity (Θ) require a larger encoder buffer. The above analysis of maximum buffer size can also be applied to the decoder buffer.

C. Adaptive Buffer Regulation for SRC

In Section V-B, we have derived the maximum encoder and decoder buffer size for video streaming with SRC. From (33) we can see that the buffer size linearly increases with the smoothing window size. In practice, the acceptable buffer size, or buffer delay, is determined by the application requirement, which is often much smaller than the maximum buffer size obtained from theoretical analysis. In this case, we need to develop a buffer regulation scheme which guarantees neither buffer overflow nor buffer underflow occurs. Note that buffer regulation and quality smoothing are two conflicting factors. In robust buffer regulation, the encoding bit rate has to be well controlled within some range specified by the buffer overflow and underflow criteria, in spite of dramatic scene change and quality variation. However, in quality smoothing rate control, the encoder has to maintain a smoothed quality change from frame to frame, in spite of large bit rate fluctuation. In this section, we propose a buffer-constrained SRC scheme which finds a good trade-off between buffer regulation and quality smoothing.

The proposed robust buffer control operates as follows: During the initial stage of transmission, let the bits in the buffer accumulate to a safe or desired buffer level, denoted by W_0. For example, one can set W_0 to be $0.5\bar{W}$ where $\bar{W}$ is the buffer size. Once the buffer level, denoted by W, reaches W_0, the network channel starts to drain the encoder buffer and transmit the bits. When W is above or below the safe level W_0, the encoder has to decrease or increase the coding bit rate according to some policy. Obviously this policy needs to consider how "urgent" the current buffer situation is and act accordingly by setting the rate change amount and speed. In addition, this policy needs to negotiate with the SRC module and try to maintain the video presentation quality as smooth as possible. Let $W_{res} = W - W_0$. If $W_{res} > 0$, we need to adjust the encoder to reduce the output bits by W_{res} during the next short period of time such that the buffer level goes back to the desired level. Specifically, we set the "short period of time" to be $0.5M$ (after being rounded to integer) frames where M is the SRC window size. Let

$$R'_T = R_T - \frac{W_{res}}{0.5M}. \tag{34}$$

The R_T in (21) is replaced by R'_T. The quantization parameter q_C and the CBR distortion level D_C obtained with R'_T are then used for SRC. This procedure implies that the encoder is trying to produce W_{res} less bits during the next $0.5M$ frames. Since the buffer control is designed as an integrated part of the SRC algorithm, the encoder will still be able to maintain a smoothed picture distortion profile with the lowpass filtering mechanism.

VI. Algorithm

In this section, we summarize the algorithm for SRC and analyze its computation and implementation complexity. The proposed SRC algorithm has the following major steps:

Step 1 *Initialization.* The first M frames of the video sequence are encoded in CBR mode. For each frame, the coding distortion is stored as $\{D_C(n)\}$. The following SRC procedure then starts from frame $M+1$.

Step 2 *Determine the target distortion level.* Suppose the current frame number is n. Its target distortion level $D_S(n)$ is obtained with (1). After motion compensation and DCT, the distribution information of the DCT coefficients are collected. Using the formula in (23), we can find the quantization parameter, denoted by q_S, such that $D_S(n) = D(n; q_S)$.

Step 3 *Encoding.* q_S is used to quantize the DCT coefficients. After entropy encoding, the actual bit rate is recorded as $R_S(n)$.

Step 4 *Estimate CBR distortion.* Using the method discussed in Sections IV-A and IV-B, specifically, (21)-(24), we can estimate the picture distortion in CBR coding mode $D_C(n)$.

Step 5 *Loop* Repeat Steps 2 to 4 until all frames are encoded.

It can be seen that, in the proposed SRC algorithm, the major computation is just to collect the distributions of the DCT coefficients. The rest of the algorithm involves only a few number of addition, multiplication, and power operations. Therefore, the algorithm has very low computational complexity and implementation cost.

VII. Experimental Results

We have implemented the proposed quality smoothing rate control algorithm in MPEG-4 video encoding [12], and tested its performance in real-time video recording and streaming. We used several TV news, movie, and sports clips for the test. In order to allow other researchers to reproduce the experimental results presented in this paper, we also used the standard MPEG video sequences, including "Football", "Flower garden", "Table tennis", "Foreman", "Coastguard" and "NBA", all short video clips. To demonstrate the performance of the SRC algorithm more efficiently, we cascade these short video clips to generate one long video clip of 1200 frames (40 seconds). All the test videos are in CIF size (352 $\times$ 288) at 30 fps (frame per second). Only I and P frames are used, and the GOP size is 60 [1]. In the following experiments, we use the TM5 bit allocation algorithm [16] for performance comparison. TM5 uses an efficient frame-level bit allocation algorithm to find the target bit rate for each video frame such that frame-to-frame quality change is smooth and the overall quality is optimized. Three algorithms to be evaluated are: TM5 bit allocation which is labeled as "without SRC"; the proposed SRC algorithm which is labeled as "with SRC"; and the buffer-constrained SRC algorithm which is labeled as "constrained SRC".

In Fig. 3, we plot the PSNR (peak signal-to-noise ratio) values of each frame encoded without SRC (dotted line) and with SRC (solid line) for the long standard video clip. It can be seen that, with the SRC algorithm, the frame-to-frame quality

[1]In case of sudden scene changes, I-frames could be used by the encoder, and therefore the actual GOP size could be dynamic and less than 60 frames.

variation has been significantly reduced, and the output video has a smoothed quality profile. Fig. 4 plots the encoding bits of each frame. As expected, the SRC algorithm has a larger variation in bit rate. As mentioned before, this is allowed in many offline and real-time video recording applications so long as the total video data storage size is met, which has been guaranteed by our theoretical analysis in Section III. However, in real-time video streaming, the buffer size has to be limited and the buffer delay has to be kept as small as possible. Using the constrained SRC algorithm discussed in Section V-C, we can achieve both robust buffer regulation and quality smoothing. Fig. 5 plots the encoder buffer level for a buffer size of 30 frames, which corresponds to 1 second of delay. It can be seen that the buffer control is very robust without buffer overflow and underflow problems. Also, the SRC algorithm is trying to take full advantage of the buffer resource to maximize the video presentation quality. Fig. 6 plots the PSNR values of each frame encoded without SRC and with the constrained SRC algorithm. We can see that the constrained SRC algorithm is still able to maintain smoothed video quality across frames while satisfying the buffer constraint. Figs. 7-10 show the results for a typical TV news clip and demonstrate a similar performance of the proposed SRC algorithm.

To evaluate the distortion smoothing performance more systematically, we use the following measure for video quality variation [1],

$$\mathcal{S}(\{D(n)\}) = \frac{1}{N-1}\sum_{n=1}^{N}|D(n) - D(n-1)|, \qquad (35)$$

where $\{D(n)\}$ is the distortion profile of the encoded video, and N is the length of the video clip. Table I lists the values of $\mathcal{S}(\{D(n)\})$ for the above two test videos, as well as for several other video clips, such as movie and TV sports clips. Here, the picture distortion $D(n)$ refers to the mean square error between the original and the reconstructed pictures [2]. We can see that SRC has dramatically reduced the picture quality variation in the encoded videos, by up to 10 times. With the buffer constraint, the quality variation measure has only been increased slightly. In our simulations, we observe that the SRC algorithm doesn't improve the average PSNR of the video sequence, and maintains similar average PSNR values as the video encoding without SRC. This is because of the low-pass filtering nature of the SRC algorithm.

TABLE I
COMPARISON OF VIDEO QUALITY VARIATION.

Video Clips	Quality variation $\mathcal{S}(\{D(n)\})$		
	Without SRC	With SRC	Constrained SRC
Standard clip	3.23	0.33	0.37
TV news	4.59	0.47	0.54
Movie clip	3.89	0.39	0.44
TV sports clip	5.10	0.55	0.60

A. Further Discussion

From the experimental results presented in the above, we can see that using the geometric lowpass filtering of R-D functions, the encoder is able to smooth out the local picture quality fluctuation while meeting the target bit rate automatically. However, the long-term quality variation still exists. This is inevitable in real-time video processing where the access of global statistics is not available. Certainly, we can increase the window size of the geometric lowpass filter to smooth out even larger scale quality fluctuation at the cost of longer buffer delay. However, from the theoretical analysis in Section III, we cannot increase the window size to be too large, otherwise, the bit rate matching will be a problem. This is because in (11), it is required that N should be sufficiently large when compared to the window size M. According to our simulation experience, suppose the length of the video sequence is T, a good choice of the window size should be less than $\frac{T}{15}$.

During the past few years, a number of algorithms have been developed for constant-quality video encoding and communication [2], [4], [18]. In non-realtime offline video encoding, two-pass bit allocation and rate control schemes can be used [2]. In real-time video encoding, the quality smoothing problem becomes more challenging because the encoder has to match its average bit rate to the available bandwidth or storage space. The quality smoothing algorithms proposed in [4], [18], unlike the geometric low-pass filtering in our SRC algorithm, do not provide an analytic model-level mechanism to guarantee the bit rate target. The major contribution of this work is the development of a quality smoothing framework using low-pass filtering of R-D functions, which achieves quality smoothing and bit rate matching simultaneously.

VIII. CONCLUSION

We have introduced the concept of low-pass filtering of rate-distortion (R-D) functions and developed the SRC algorithm for real-time video recording and streaming applications. Both the theoretical analysis and experimental results have shown that the SRC algorithm is able to meet the target bit rate accurately while maintaining a smoothed video presentation quality. For real-time network video streaming, we have also integrated the buffer control into the SRC framework. Our experimental results show that the robust buffer regulation can be achieved with negligible degradation in quality smoothing performance of the SRC algorithm. The proposed SRC algorithm has direct application in quality control and performance optimization in real-time video encoding and streaming system design.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] L.-J. Lin and A. Ortega, "Bit-rate control using piecewise approximated rate-distortion characteristics," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 4, 00. 446-459, Aug. 1998.

[2] For a better evaluation of video quality, some perceptual video quality measures, such as the JND measure [17], could be considered. However, these perceptual measures are often very complicated and not as mathematically tractable as the PSNR formulation

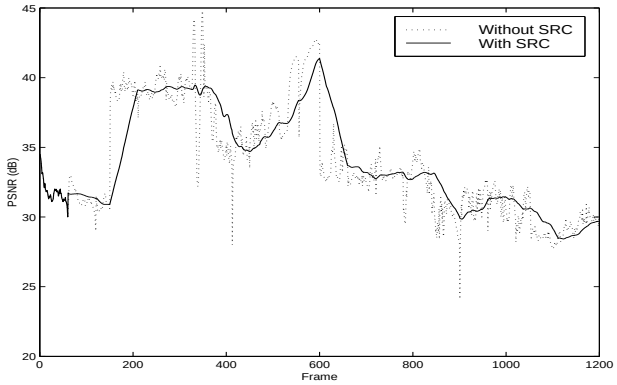

Fig. 3. PSNR of each frame encoded without SRC (dotted line) and with SRC (solid line) for the standard video clip.

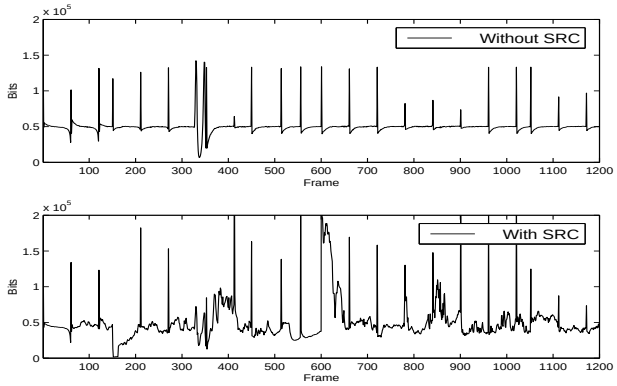

Fig. 4. Output bits of each frame encoded without SRC (top) and with SRC (bottom) for the standard video clip.

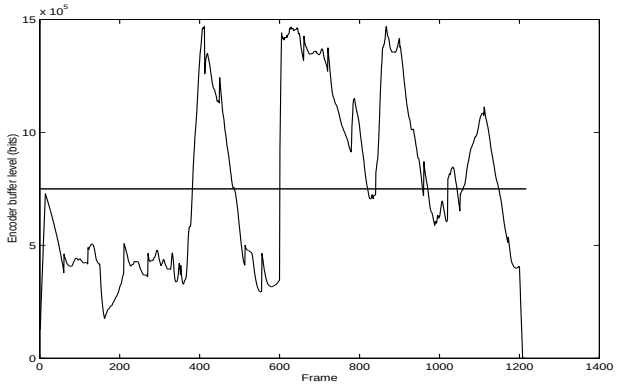

Fig. 5. Encoder buffer level at each frame. The buffer size is 1500 Kbits which corresponds to a buffer delay of 1 second for the standard video clip.

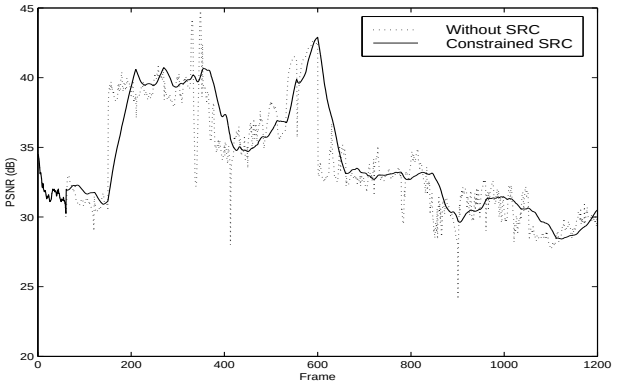

Fig. 6. PSNR of each frame encoding without SRC and with buffer-constrained SRC for the standard video clip.

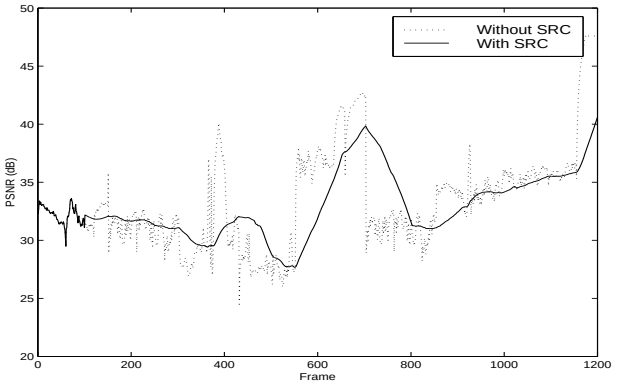

Fig. 7. PSNR of each frame encoded without SRC (dotted line) and with SRC (solid line) for the TV news clip.

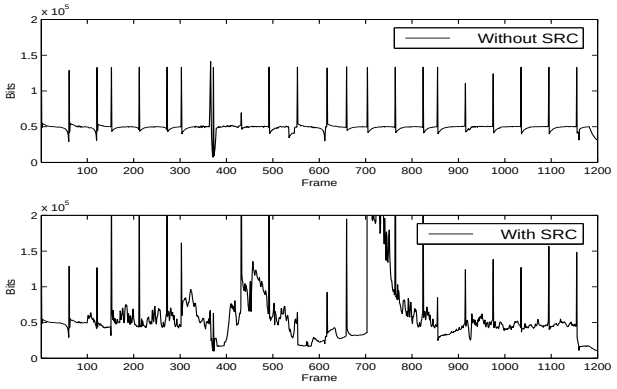

Fig. 8. Output bits of each frame encoded without SRC (top) and with SRC (bottom) for the TV news clip.

[2] Y. Yu, J. Zhou, Y. Wang, and C. W. Chen, "A novel two-pass VBR coding algorithm for fixed-size storage application", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 345 -356, March 2001.

[3] Windows Media Encoder 9 Series, http: // www.microsoft.com / windows / windowsmedia / 9series/.

[4] B. Xie and W. Zeng, "Seqeunce-based rate control for contant quality video," *Proceedings of International Conference on Image Processing*, vol. I, pp. 77-80, Rochester, NY, September 2002.

[5] "MPEG-2 Video Test Model 5," *ISO/IEC JTC1/SC29/WG11 MPEG93/457*, April 1993.

[6] ITU-T, "Video coding for low bit rate communications," *ITU-T Recommendation H.263*, version 1, version 2, January 1998.

[7] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 19–31, February 1997.

[8] Z. He, Y. Kim, and S. K. Mitra, "A novel linear source model and a unified rate control algorithm for H.263/MPEG-2/MPEG-4," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, May 2001.

[9] C.-Y. Hsu, A. Ortega and A. R. Reibman, "Joint Selection of Source and Channel Rate for VBR Video Transmission under ATM Policing Constraints," *IEEE Journal on Sel. Areas in Communications, Special Issue on Real-Time Video Services in Multimedia Networks*, Vol. 15, No. 6, pp. 1016-1028, August 1997.

[10] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE. Trans. on Inform. Theory*, vol. IT-14, pp. 676–683, September 1968.

[11] T. Berger, *Rate Distortion Theory,* Prentice Hall, Englewood Cliffs, NJ,

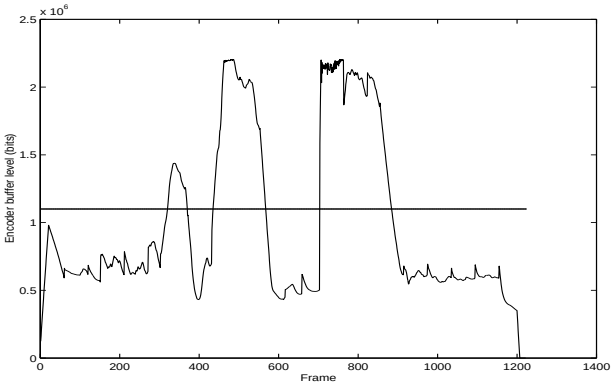

Fig. 9. Encoder buffer level at each frame. The buffer size is 1500 Kbits which corresponds to a buffer delay of 1.5 second for the TV news clip.

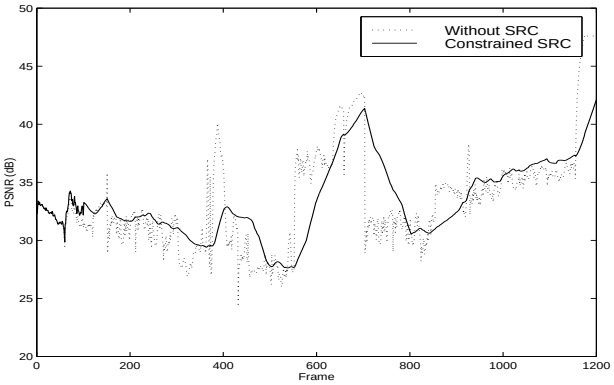

Fig. 10. PSNR of each frame encoding without SRC and with buffer-constrained SRC for the TV news clip.

1984.

[12] MoMuSys codec, "MPEG4 verification model version 7.0," *ISO / IEC JTC1 / SC29 / WG11 Coding of Moving Pictures and Associated Audio MPEG97*, Bristol, U.K., March 1997.

[13] T. Chiang, Y. -Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.7, pp. 246 – 250, February 1997.

[14] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp. 172 – 185, February 1999.

[15] Z. He, Y. Kim, S. K. Mitra, "A linear source model and a unified rate control algorithm for H.263 / MPEG-2 / MPEG-4," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, May 2001.

[16] D. LeGall, "MPEG: A video compression standard for multimedia application," *Commun. ACM*, vol. 34, pp. 46–58, April 1991.

[17] J. Lubin and D. Fibush, "Sarnoff JND vision model," T1A1.5 Working Group Document No. 97-612, ANSI T1 Standards Committee, 1997.

[18] I. Dalgic and F. A. Tobagi, "Constant quality video encoding" Communications," *ICC 95 Seattle, Gateway to Globalization, 1995 IEEE International Conference on* vol. 2, pp. 1255 -1261, June 1995.

PLACE PHOTO HERE

Zhihai He received the B.S. degree from Beijing Normal University, Beijing, China, and the M.S. degree from Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing, China, in 1994 and 1997 respectively, both in mathematics, and the Ph.D. degree from University of California, Santa Barbara, CA, in 2001, in electrical engineering. In 2001, he joined Sarnoff Corporation, Princeton, NJ, as a Member of Technical Staff. In 2003, he joined the Department of Electrical and Computer Engineering, University of Missouri, Columbia, as an assistant professor. He received the 2002 IEEE Transactions on Circuits and Systems for Video Technology Best Paper Award, and the SPIE VCIP Young Investigator Award in 2004. His current research interests include image/video processing and compression, network transmission, wireless communication, computer vision analysis, sensor network, and embedded system design. He is a member of the Visual Signal Processing and Communication Technical Committee of the IEEE Circuits and Systems Society, and serves as Technical Program Committee member or session chair of several international conferences.

PLACE PHOTO HERE

Wenjun Zeng (S94-M97-SM03) received the B.E. degree from Tsinghua University, Beijing, China, in 1990, the M.S. degree from the University of Notre Dame in 1993, and the Ph.D. degree from Princeton University in 1997, all in electrical engineering.

He has been an Associate Professor with the Computer Science Department of University of Missouri, Columbia, MO since Aug. 2003. He worked at Matsushita Information Technology Lab, Panasonic Technologies Inc., Princeton, in the summer of 1995, and at Multimedia Communication Lab, Bell Laboratories, Murray Hill, NJ, in the summer of 1996. From 1997 to 2000, he was with Sharp Labs of America, Camas, WA. He was with PacketVideo Corporation, San Diego, from December 2000 to August 2003, where he was leading R&D projects on wireless multimedia streaming, encoder quality optimization, and digital rights management. His current research interests include multimedia communications and networking, content and network security, and wireless multimedia. He has been an active contributor to the JPEG 2000 image coding standard and the MPEG4 IPMP Extension standard, where four of his proposals have been adopted into the standards. He has been awarded 9 patents.

Dr. Zeng has served as a Technical Program Committee Member, Special Session Chair, and Panel Session Organizer for several IEEE international conferences. He was the Lead Guest Editor of IEEE Transactions on Multimedia Special Issue on Streaming Media published in April 2004. He is the Technical Program Co-Chair of the Multimedia Communications and Home Networking Symposium, IEEE International Conference on Communications, 2005. He is a member of the IEEE COMSOC Multimedia Communications Technical Committee.

PLACE PHOTO HERE

Chang Wen Chen