

# Rate-Reduction Transcoding Design for Video Streaming Applications

Jianfei Cai<sup>a</sup> and Zhihai He<sup>b</sup> and Chang Wen Chen<sup>b</sup>

<sup>a</sup>Univ. of Missouri-Columbia, Dept. of Electrical Engineering, Columbia, MO 65211

<sup>b</sup>Sarnoff Corporation, 201 Washington Road, Princeton, NJ 08543

## ABSTRACT

Due to the heterogeneity problem existing in Internet and the diversity of users' requirements, rate-reduction transcoding is necessary in many video streaming applications. In this paper, we develop a novel rate-reduction transcoder. Specifically, by taking advantage of the pre-generated feature information of video sequences, in the transcoding stage, we are able to efficiently allocate bits among video frames. In addition, we simplify the transcoder architecture so that the transcoding can be implemented in a fast way. Preliminary results demonstrate that the proposed transcoder is able to achieve not only the reduced average distortion but also much smooth video quality.

## 1. INTRODUCTION

Video streaming services, such as video-on-demand, digital library and non-interactive distance learning, are becoming more and more popular in our daily lives. Unlike text or image, video sequences typically have huge volume in data size. For example, a common TV resolution RGB video ( $576 \times 720$  pixels) with 30 frames/second requires about 300 Mbps bandwidth. It is clear that digital video data, in its original format, are too voluminous for transmission or storage. A certain degree compress is desired, which depends on the target bit rate, i.e., the bandwidth of the channel. In a typical video streaming system, video sequences are encoded in advanced and stored in the server. Users can access the server through an access networks.

Since different users may have different video decoders and display devices, and they may access video servers through different access networks, for a video sequence, only one bitstream coded at a certain bit rate and stored at the servers cannot satisfied the requirement of different users. Generally, there are three solutions. One is to store many bitstreams for one video sequence. Each bitstream is coded at different formats or different bit rates. When a user requests to access the video sequence, the server can send the bitstream which is closet to the user's requirements. However, this method is rarely used because it will cost a lot of storage spaces in the video servers and the chosen bitstream may not satisfy the user's requirement exactly. Another solution is to

use multiresolution or scalable coding such as MPEG-4 FGS coding.<sup>1</sup> An obvious disadvantage of FGS coding is that the degradation becomes significant when the base layer is at low bit rate. The third solution is that, for each video sequence, only one bitstream coded at high quality is stored in the video server. When a user requests an access to a video sequence, real-time video transcoding is performed so that the transcoded bitstream can match the user's requirement. Since video transcoding does not require extra storage spaces and is very flexible, it is widely adopted in practical video streaming applications.

Many transcoding schemes<sup>2-8</sup> have been developed in the past few years. Generally, there are two types of video transcoding: format transcoding and rate-reduction transcoding. Format transcoding represents transcoding between different video formats such as between MPEG-4 and H.263. Rate-reduction transcoding represents transcoding from higher bit rate to lower bit rate. The rate-reduction transcoding can be further classified into two categories: pixel-domain transcoding and DCT-domain transcoding. The performance of pixel-domain transcoding is better than DCT-domain transcoding while the complexity of pixel-domain transcoding is much higher.

In this research, we focus on rate-reduction transcoding. we proposed a novel transcoding scheme based on our previous work – a low bit rate offline video coding scheme<sup>9</sup> for video streaming applications. Specifically, the proposed transcoding scheme exploits the feature information of video sequences, pre-generated at the server, so that better performance can be achieved at the transcoding stage. Moreover, we also proposed a novel DCT-domain transcoding architecture, which is able to greatly reduce the transcoding complexity.

The paper is organized as follows. Section 2 introduces our previous work – a low bit rate offline video coding scheme. Section 3 describes the transcoding rate control. Section 4 presents the proposed transcoding architecture. Section 5 gives the experimental results. Finally, section 6 concludes this paper.

## 2. LOW BIT RATE OFFLINE VIDEO CODING

In our previous work,<sup>9</sup> by notice that current rate control schemes in video coding standards do not have efficient frame-level bit allocation, we have proposed a rate control scheme based on optimal frame-level bit allocation for low bit rate offline video coding. Ideally, in order to achieve optimal frame-level bit allocation, it is desired to generate R-D functions of all the video frames in an entire video sequence. For standard video coding, rate  $R$  and distortion  $D$  largely depend on the quantization parameter  $q$ . Therefore, R-D functions can be expressed as  $R(q)$  and  $D(q)$  functions. if the R-D function for each frame is independent, we can easily generate the R-D functions by coding the video sequence multiple times. However, for standard video coding, the R-D function

of each frame is not independent due to motion compensation. This makes the generation of the R-D functions for all the video frames impossible.

Recently, a novel R-D model is developed in<sup>10</sup> for DCT-based video coding. In that model, the source coding rate  $R_i$  and the distortion  $D_i$  of a frame  $i$  are considered as functions of  $\rho_i$  which is the percentage of zero among the quantized DCT coefficients of the frame  $i$ . Specifically, the rate model can be written as

$$R_i(\rho_i) = \theta_i(1 - \rho_i)N_p + H_i, \quad (1)$$

where  $\theta_i$  is a constant,  $N_p$  is the number of pixels in a frame, and  $H_i$  refers to the number of bits for the header information and the motion vectors for the frame  $i$ . The distortion model can be written as

$$D_i(\rho_i) = \sigma_i^2 e^{-\alpha_i(1-\rho_i)}, \quad (2)$$

where  $\sigma_i$  is the standard deviation of the frame  $i$ , and  $\alpha_i$  is a constant.

We adopt these R-D models for the frame-level bit allocation. Suppose there are  $L$  frames, based on Eqns. (1) (2), the optimum frame-level bit allocation can be formulated as

$$\min_{\rho_i} \sum_{i=1}^L \sigma_i^2 e^{-\alpha_i(1-\rho_i)}, \quad (3)$$

$$s.t. \quad N_p \sum_{i=1}^L \theta_i(1 - \rho_i) + \sum_{i=1}^L H_i = \frac{R_s L}{R_f}, \quad (4)$$

where  $\frac{R_s L}{R_f}$  is the total number of bits available for the  $L$  frames. With the Langrange multiplier, we can convert this constrained minimization problem into an unconstrained problem, i.e.,

$$\min_{\rho_i} \sum_{i=1}^L \sigma_i^2 e^{-\alpha_i(1-\rho_i)} + \lambda [N_p \sum_{i=1}^L \theta_i(1 - \rho_i) + \sum_{i=1}^L H_i - \frac{R_s L}{R_f}]. \quad (5)$$

By solving this minimization problem, the optimal number of bits for a frame  $i$  can be calculated as

$$R_i = \left[ \frac{R_s L}{R_f} - \sum_{i=1}^L H_i - N_p \sum_{i=1}^L \eta_i \log\left(\frac{\sigma_i^2}{\eta_i}\right) \right] \frac{\eta_i}{\sum_{i=1}^L \eta_i} + N_p \eta_i \log\left(\frac{\sigma_i^2}{\eta_i}\right) + H_i, \quad (6)$$

where  $\eta_i = \frac{\theta_i}{\alpha_i}$ . As shown in Eqn. (6), in order to optimally allocate bits among frames, we need to collect the feature information of each frame, including  $\sigma_i$ ,  $\theta_i$ ,  $\alpha_i$  and  $H_i$ ,  $i = 1, \dots, L$ . Since, for video streaming applications, video sequences are usually available in the video servers in advance, we can pre-encode the video sequences once to generate the feature information. Based on the pre-generated feature information, a rate control algorithm has been developed in.<sup>9</sup>

### 3. RATE CONTROL IN TRANSCODING

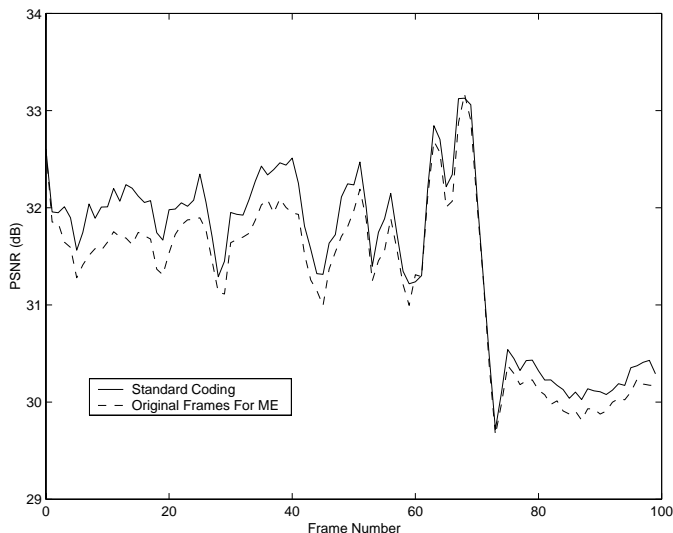
In the view of rate control, the task for rate-reduction transcoding can be represented as

$$Video(R_s^0) \rightarrow Video(R_s), \quad R_s < R_s^0, \quad (7)$$

where  $R_s^0$  is the coding rate for original video bitstreams and  $R_s$  is the new coding rate. Since the feature information of original video sequences has already been generated in the server, we can use this feature information for transcoding. Although the feature information of the reconstructed video will be different from that of the original video, the difference is assumed small because the original video is coded at high quality. Therefore, we can use the same rate control algorithm, proposed in our previous work,<sup>9</sup> to optimally allocate bits among frames under new bandwidth constraints.

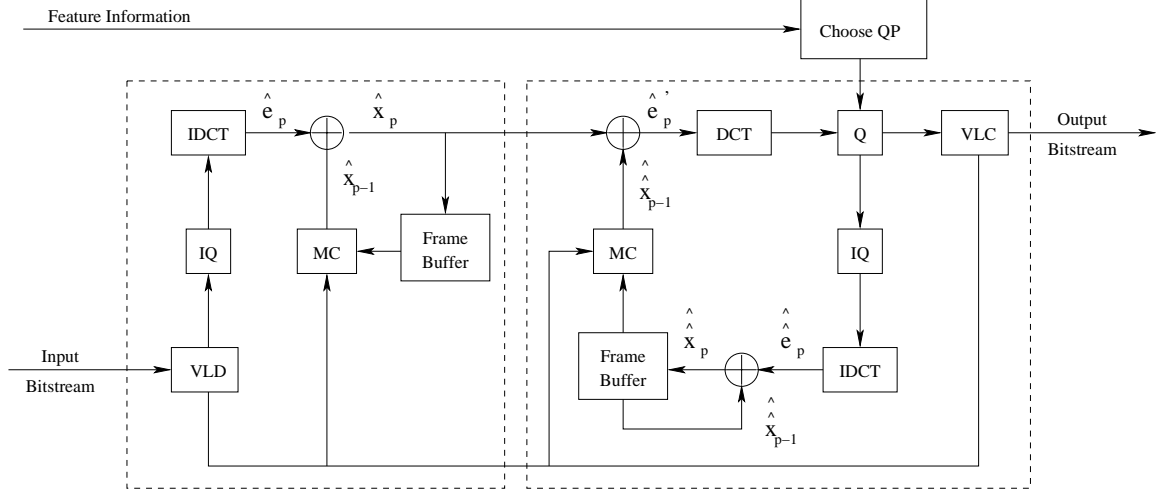
### 4. TRANSCODING ARCHITECTURE

Since transcoding is usually performed at real-time, it is desired to have a low complexity transcoder. A common transcoder is a decoder followed by an encoder. Such a transcoding architecture is too complicate. We observed that using original frames for motion estimation (ME) while still using reconstructed frames for motion compensation (MC) will only cause little performance degradation and have no drift problem. Fig. 1 shows such an example. Therefore, we directly applied the original motion vectors in transcoding and there is no need for motion estimation. The modified



**Figure 1.** The PSNR results of coding QCIF Foreman sequence using standard MPEG-4 codec with no rate control and a fix quantization parameter 12.

transcoder is shown in Fig. 2. Although the performance will degrade a little, the computation will be greatly reduced since ME is usually considered as the most expensive portion in the sense of computation cost.



**Figure 2:** A modified transcoder.

Through some mathematic manipulation, we are able to further reduce the transcoding complexity. As shown in Fig. 2, the new MC residue for pixel  $(m, n)$  can be expressed as

$$\begin{aligned}\hat{e}'_p(m, n) &= \hat{x}_p(m, n) - \hat{x}_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n)) \\ &= \hat{e}_p(m, n) + \hat{x}_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n)) - \hat{x}_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n))\end{aligned}\quad (8)$$

where  $\hat{e}_p(m, n)$  denotes the first reconstructed residue of pixel  $(m, n)$  of the  $p$ -th frame,  $\hat{x}_{p-1}(u, v)$  and  $\hat{\hat{x}}_{p-1}(u, v)$  are the first and the second reconstructed pixel  $(u, v)$  of the  $(p - 1)$ -th frame, and  $d_p^v(m, n)$  and  $d_p^h(m, n)$  denote the vertical and horizontal motion vector for pixel  $(m, n)$  of the  $p$ -th frame. It is clear that two frame buffers are needed in order to store  $\hat{x}_{p-1}$  and  $\hat{\hat{x}}_{p-1}$ . If we define  $y_p$  as the difference between two reconstruction, that is

$$y_p(m, n) = \hat{x}_p(m, n) - \hat{\hat{x}}_p(m, n), \quad (9)$$

we can further express  $\hat{e}'_p$  as

$$\hat{e}'_p(m, n) = \hat{e}_p(m, n) + y_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n)). \quad (10)$$

Since  $\hat{x}_p(m, n) = \hat{e}_p(m, n) + \hat{x}_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n))$  and  $\hat{\hat{x}}_p(m, n) = \hat{e}_p(m, n) + \hat{\hat{x}}_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n))$ , where  $\hat{e}_p(m, n)$  denotes the second reconstructed residue of pixel  $(m, n)$  of the  $p$ -th frame,  $y_p$  can be updated as

$$y_p = \hat{e}_p(m, n) - \hat{\hat{e}}_p(m, n) + y_{p-1}(m - d_p^v(m, n), n - d_p^h(m, n)), \quad (11)$$

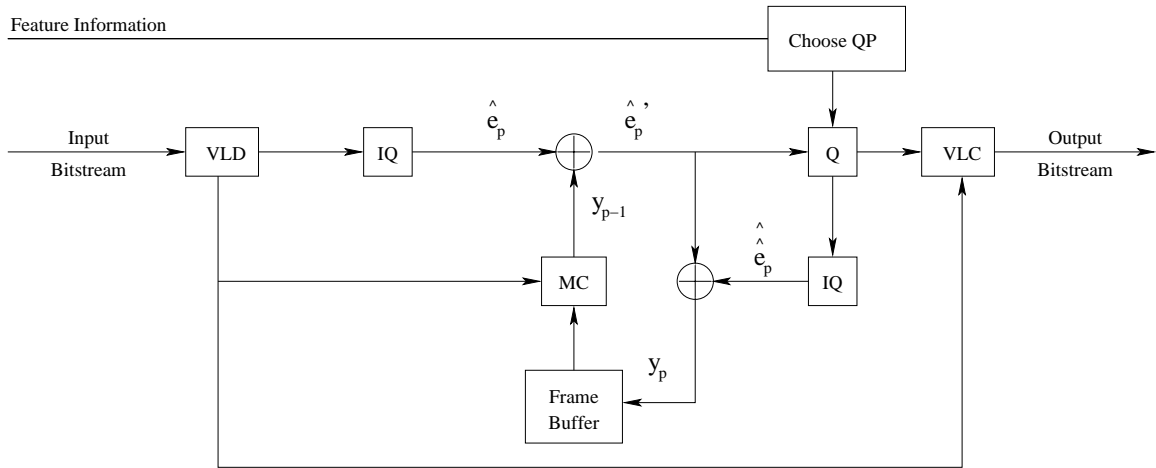
with initial value

$$y_0(m, n) = \hat{x}_0(m, n) - \hat{\hat{x}}_0(m, n). \quad (12)$$

Considering the relationship in Eqn. (8),  $y_p$  can be further simplified as

$$y_p(m, n) = \hat{e}'_p(m, n) - \hat{\hat{e}}_p(m, n). \quad (13)$$

In this way, there is no need to generate and store the reconstructed frames  $\hat{x}$  and  $\hat{\hat{x}}$ , and only  $y_p$  is needed to be stored. Because DCT transform is essentially a linear transform, the derivation above can be extended to DCT domain. Fig. 3 shows the simplified transcoding system, where only one memory and only one MC unit are needed.



**Figure 3:** A simplified transcoder.

## 5. EXPERIMENTAL RESULTS

In this section, we perform experiments based on the standard H.263 codec to illustrate the effectiveness of the proposed transcoding. Similar results can be obtained by using other video coding standards. The experiments are performed on two QCIF format video sequences. One is 300 frames of “Foreman”, which contains large facial movements and camera panning at the end. The other is a video sequence containing multiple scenes: 100 frames of “Foreman”, fast motion, 100 frames of “Mother & Daughter”, slow motion, and 100 frames of “Coastguard”, fast motion. The original video sequences are coded by H.263 coder at 10 fps and 128 kbps. For other coding rates, similar results can be obtained.

We compare our proposed transcoding scheme with the HIST transcoding scheme. In our proposed transcoding scheme, the original video sequences are first coded by the previous proposed

offline video encoder<sup>9</sup> and then the compressed bitstreams are transcoded to low bit rates by our proposed transcoder. In the HIST transcoding scheme, the original video sequences are coded by the rate control scheme proposed in,<sup>10</sup> which is termed as HIST, and then the compressed bitstreams are fully decoded and re-encoded at low bit rates by the HIST rate control scheme. As shown in,<sup>10</sup> HIST is a rate control scheme which is able to achieve better rate control performance than TMN8 in H.263. Generally speaking, HIST is a MB-level rate control scheme, and the frame-level bit allocation method adopted in HIST is the same as that in TMN8. Since HIST is adopted for MB-level rate control in the rate control scheme<sup>9</sup> used in our proposed transcoding system, the comparison between the rate control scheme<sup>9</sup> and HIST can be considered as the comparison between the optimal frame-level bit allocation method<sup>9</sup> and the frame-level bit allocation method in TMN8. Due to the limitation of real-time applications, the frame-level bit allocation in TMN8 is very simple, i.e., equally assigning bits among video frames. Therefore, as demonstrated in,<sup>9</sup> the optimal frame-level bit allocation method can achieve much better performance. In the following experiments, we will demonstrate that the optimal frame-level bit allocation can also achieve good performance for transcoding.

### 5.1. Performance Parameters

The performance parameters include "Average Distortion", "Distortion STD", "BW Diff.", "BW Error", "Buffer Size" and "Pre-loading Time". "Average Distortion" ( $\bar{D}$ ) denotes the average MSE of a frame, which is calculated as

$$\bar{D} = 10 \log_{10} \left( \frac{1}{L} \sum_{i=1}^L D_i \right). \quad (14)$$

Notice that although the average PSNR is widely used in literature, we find it is not appropriate to represent the average quality of a video sequence. This is because the maximal average PSNR does not correspond to the minimal average distortion due to the logarithm function while our target is to minimize the average distortion. Therefore, in this research, we use the average distortion instead of the average PSNR to measure the average quality of a video sequence while we still use PSNR to measure the quality of each individual frame. "Distortion STD" ( $\sigma_D$ ) denotes the standard deviation of the frame distortions, which is calculated as

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

"BW Diff." ( $\Delta BW$ ) denotes the difference between the total used bits and the total available bandwidth, which is calculated as

$$\Delta BW = \sum_{i=1}^L R_i - \frac{R_s L}{R_f}. \quad (16)$$

"BW Error" ( $BW_e$ ) is calculated as

$$BW_e = \frac{|\Delta BW|}{R_s L / R_f}. \quad (17)$$

"Buffer Size" and "Pre-loading Time" denote the required buffer size and the required pre-loading time in order to guarantee no buffer underflow and overflow under a constant channel transmission rate.

## 5.2. Simulation Results

Table 1 shows the transcoding results at different bit rates. As shown in this table, the reduction of the average distortion is from 0.24 dB to 1.19 dB, and the reduction of  $\sigma_D$  is from 3.51 dB to 10.53 dB. The lower STD indicates the subjective performance of the proposed transcoding scheme is more consistent. This can be shown more clearly in Fig. 4. Comparing with the bandwidth control error in the offline video coding,<sup>9</sup> the bandwidth control error in the proposed transcoding system is a little bit larger. This is because the feature information of the original video sequences is used in the proposed transcoding system. Also shown in Table 1, both transcoding scheme have low  $BW_e$ , no more than 0.25%. Although the proposed scheme requires larger buffer size and longer pre-loading time, the required buffer size is only several hundred kilobits and the required pre-loading time is less than 1.5 s, which is reasonable for video streaming applications. Although in some cases, with the increasing of the length of the video sequence, the required buffer size may become much larger and the pre-loading time may become much longer, we can solve this problem by introducing a window, which contains a certain number of frames, and only performing the optimal frame-level bit allocation among the frames within each window. Also, notice that in the previous statement, we state that original video sequences are pre-encoded at high-quality. However, in the experiments, we show that good performance can be achieved in transcoding video sequences pre-encoded at 128 kbps, which is not a high bit rate. This indicates that, for our proposed transcoding, the original video sequences do not have to be coded at high quality.

## 6. CONCLUSION

In this work, we have developed a high-performance and low-complexity transcoder for video streaming applications. Based on the pre-generated feature information, at the transcoding stage, we are able to efficiently allocate bits among video frames. Such frame-level bit allocation can result in 0.24 dB to 1.19 dB reduction for the average distortion and, more outstandingly, it can achieve at least 3.5 dB reduction for the standard deviation. It indicates that the proposed transcoding can achieve not only better but also much smoother visual quality. On the other hand, by employing original frames for ME, we are able to simplify the transcoder architecture so that no ME and

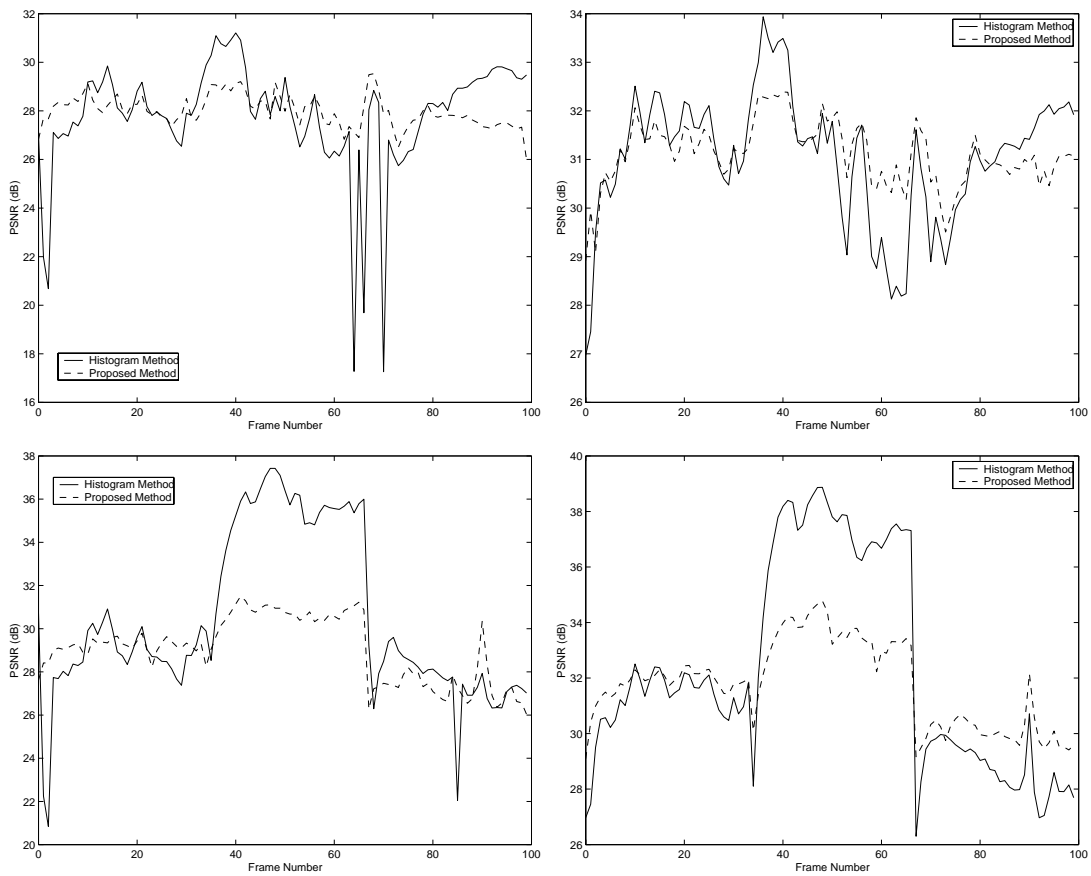
**Table 1:** The performance of transcoding from 128 kbps to low bit rate.

Channel Rate (kbps)	RC Scheme	Average Distortion (dB)	Distortion STD (dB)	BW Diff. (bits)	BW Error (%)	Buffer Size (kbits)	Pre-loding Time (second)
video sequence 1							
32	HIST	21.38	22.47	16	0.005	10.08	0.32
	Proposed	20.19	11.94	272	0.085	34.24	1.07
64	HIST	17.32	12.75	-48	0.008	10.08	0.16
	Proposed	17.08	9.16	40	0.006	49.78	0.53
video sequence 2							
32	HIST	20.27	19.35	344	0.1	10.08	0.32
	Proposed	19.38	14.87	800	0.25	61.99	0.95
64	HIST	17.25	15.57	32	0.005	10.08	0.15
	Proposed	16.67	12.06	96	0.015	130.10	0.58

only one MC unit and one frame buffer unit are needed in the proposed transcoder, which greatly reduces the transcoding complexity.

## REFERENCES

1. 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.
2. J. Youn, M. Sun, and C. Lin, "Motion vector refinement for high-performance transcoding," *IEEE Trans. on Multimedia*, pp. 30–40, March 1999.
3. P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 953–967, Dec. 1998.
4. H. Sun, W. Kwok, and J. Zdepski, "Architecture for mpeg compressed bitstream scaling," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 191–199, April 1996.
5. B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video down-scaling," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 929–936, April 1996.
6. O. Werner, "Requantization for transcoding of mpeg-2 intraframes," *IEEE Trans. on Image Processing*, pp. 179–191, Feb. 1999.
7. T. Shanableh and M. Ghanbari, "Transcoding architectures for DCT-domain heterogeneous video transcoding," in *Proceedings of IEEE ICIP2001*, pp. 433–436, 2001.
8. K. Seo and J. kim, "Fast motion vector refinement for MPEG-1 to MPEG-4 transcoding with spatial down-sampling in DCT domain," in *Proceedings of IEEE ICIP2001*, pp. 469–472, 2001.



**Figure 4.** The PSNR performance of transcoding from 128 kbps to low bit rate. Left: 32 kbps. Right: 64 kbps. Top: for sequence 1. Bottom: for sequence 2.

9. J. Cai, Z. He, and C. W. Chen, "Optimal bit allocation for low bit rate video streaming applications," in *submitted to IEEE ICIP 2002*, Dec. 2001.
10. Z. He, Y. Kim, and S. K. Mitra, "Low-delay rate control for DCT video coding via  $\rho$ -domain source modeling," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 928–940, Aug. 2001.