TRANSCODING TO FGS STREAMS FROM H.264/AVC HIERARCHICAL B-PICTURES

Huifeng Shen[*1], *Xiaoyan Sun*[2], *Feng Wu*[2],*Houqiang Li*[3], *Shipeng Li*[2]
[1,3]University of Science & Technology of China, [2]Microsoft Research Asia
[1]shenhf@mail.ustc.edu.cn, [2]{xysun, fengwu, spli}@microsoft.com, [3]lihq@ustc.edu.cn

ABSTRACT

This paper presents a transcoder which transcodes to FGS streams from H.264/AVC hierarchical B-pictures. First, the DCT-domain architecture is designed for fast FGS transcoding. Then, we propose a mode decision method in DCT domain to achieve a trade-off between the performances at low bit-rate and high bit-rate. Experimental results demonstrated that our method can improve the coding performance up to 1 dB at low rate and only lose at worst 0.5 dB at high rate.

Index Terms— video signal processing, video coding

1. INTRODUCTION

In universal multimedia access (UMA), different devices have different access abilities to the Internet and moreover, the networks also have different channel characteristics such as bit-rates and bit error rate at different times and different environments. Rate adaptation through transcoding [1][2] is one of the most popular methods to meet various requirements of networks and devices. However, since multiple versions of transcoding always have to be performed at the same time, these systems usually suffer from complexity problems.

Besides transcoding, video rate adaptation can also be achieved by scalable video coding [3][4]. Bit-rate adjustment is easier to achieve on scalable streams by cutting at expected points. The scalable video standard which is developing, Scalable Video Coding (SVC) [4] as extension of H.264/AVC, can provide three mechanisms of scalabilities, that is, SNR scalability, spatial scalability, and temporal scalability. It is a promising scheme which may be deployed extensively since it can achieve comparable performance with the corresponding non-scalable video scheme. However, as most of current compressed video streams are coded as non-scalable coding schemes, a transcoder that converts non-scalable video to SVC-coded video is needed to utilize the advantages of scalable video streams on rate adaptation.

On the other hand, single-layer coding schemes using hierarchical-B (H-B) structure have drawn great attention [6]. H-B represents such a coding structure that uses bidirectional predictive pictures (B-pictures) as references within one group of pictures (GOP). A typical H-B structure is shown in Figure 1. In general, bidirectional prediction can get more accurate prediction than unidirectional prediction. Thus H-B coding structure can achieve better performance on coding efficiency than traditional IBBP structure [6]. In fact, H-B structure is already supported by H.264/AVC main profile [5].

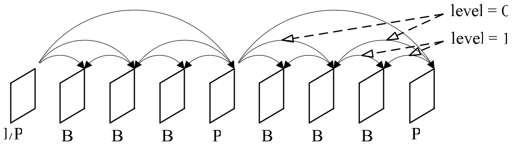

Figure 1. A typical hierarchical-B coding structure. The solid lines with arrows mean the prediction directions.

In this paper, we address the problem in which FGS-coded streams are transcoded from H.264/AVC hierarchical B-pictures. To our best knowledge, little work has addressed this problem so far. In this paper, we will first review the FGS scheme in the current SVC Joint Draft (JD), and then design a corresponding DCT-domain transcoder for FGS streams from H264/AVC coded hierarchical B-pictures. Then a rate-distortion optimal mode decision in DCT domain is introduced to achieve a good trade-off between the performances at low bit-rate and at high bit-rate. Our experimental results demonstrate that our method can improve the coding performance up to 1 dB in terms of PSNR at low rate and only lose at worst 0.5 dB at high rate. Our main contribution in this work lies on the first investigation on FGS transcoding from hierarchical B-pictures and applying our proposed mode decision method in FGS transcoding in DCT domain.

The rest of the paper is organized as follows: Section 2 reviews the FGS scheme in the current SVC JD and then a DCT-domain transcoder for FGS stream is designed; in Section 3, an RD optimal mode decision method in DCT domain is described; experimental results are presented in Section 4 and the conclusions are provided in Section 5.

2. FGS OVERVIEW AND TRANSCODER DESIGN

In this section, we will make an overview of the FGS scheme in the current SVC JD [4], and then a DCT-domain transcoder for this scheme is proposed.

[*] The work was done while the author is with Microsoft Research Asia

1-4244-0481-9/06/$20.00 ©2006 IEEE

2.1. Overview of FGS

In the current SVC JD, the fine granular scalability (FGS) is achieved by encoding successive refinement of the residual signals. This refinement is achieved by repeatedly decreasing the quantization step sizes, as illustrated in Figure 2. In the enhancement layers, with the help of the entropy coding method which is akin to the bit-plane coding, the FGS streams can be truncated at an arbitrary point. In the B-picture FGS coding scheme, for each B-picture, there are two different prediction pictures of different qualities, base-layer prediction and enhancement-layer prediction, which come from different-quality references but the same motion and mode information.

However, as shown in Figure 2, the FGS encoder involves as many DCT/IDCT pairs as the FGS layers, which is an obstacle in the case of transcoding because of the computational complexity.

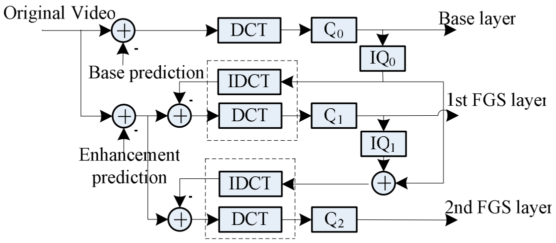

Figure 2. The diagram of the FGS encoder. Though transform and quantization in H.264/AVC is different from the conventional ones, here we still use “DCT”, “IDCT”, “Q” and “IQ” to denote transform, inverse transform, quantization and inverse quantization for convenience.

2.2. Transcoding to FGS in DCT Domain

Based on the assumption that the DCT and IDCT operations are both linear transforms, the refinement in the enhancement layers can also perform in DCT domain through merging DCT/IDCT pairs. Specially, in the case of transcoding, the quality refinement can be achieved by directly quantizing the input transform coefficients with decreasing quantization steps, as illustrated in Figure 3. In the architecture of DCT-domain transcoding, we have omitted the difference of the two references between the front-encoder and the end-decoder. In other words, we perform open-loop transcoding, which would surely bring on drifting error. However, because of the GOP structure, the drifting error within the current GOP will not propagate to the next GOP and it is not as severe as that in the IPPP structure.

However, in the conventional DCT-domain transcoding schemes, the input motion and mode information are always directly used for the output streams. Since the input streams are usually of high bit-rate and high quality, the corresponding motion and mode information are only fit for high bit-rate. However, FGS streams have to serve a wild range of bit-rates. So when the input motion information is used directly in transcoding for FGS, it would badly decrease the performance when the scalable stream is cut at low bit-rate. In the case of H-B structure, this problem is more severe since more bits are spent on motion information for better prediction, compared with that of P-pictures. In this paper, a rate-distortion (RD) optimal mode decision is used to achieve a trade-off between the performances at low and high bit-rate, while the transcoding process is still performed in DCT domain. The mode decision is described in detail below.

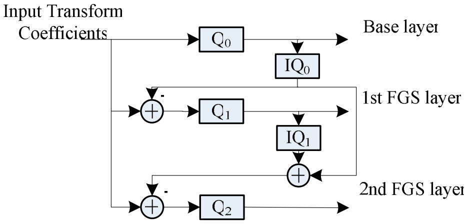

Figure 3. The transcoder architecture for FGS in DCT domain.

3. MODE DECISION IN DCT DOMAIN

In this section, an RD optimal mode decision for FGS transcoding in DCT domain is introduced.

3.1. Rate-Distortion Model in DCT Domain

In the conventional rate-distortion optimal mode decision scheme [7], the final mode and its corresponding motion are selected when minimizing

$$J = D_{total} + \lambda \times R_{total}, \qquad (1)$$

where D_{total} means the distortion resulted from the reconstructed macroblock, R_{total} means the total bits spent on coding the macroblock, and λ corresponds to the Lagrange multiplier. However, it is hard to directly use this scheme in our FGS transcoding since the pictures can not be completely reconstructed in DCT domain. Therefore, the rate-distortion model used in DCT domain needs to be investigated.

In general, the coded bits can be separated to two parts: motion bits and texture bits, i.e.

$$R_{total} = R_{texture} + R_{motion}. \qquad (2)$$

The texture bits denote the bits which are used to code the residual signals while the motion bits include the bits spent in coding the modes and motion vectors. Usually, motion information at high bit-rate needs much more bits than those at low bit-rate. So in the case of transcoding to FGS, the motion bits will cost an undue proportion at low bit-rate when the input motion information is directly used in FGS. Therefore, it is a demand to properly reduce the amount of motion bits to satisfy a wild range of bit-rates.

However, besides the distortion due to texture quantization, the alteration of motion vectors to reduce the motion bits in DCT domain would cause extra distortion, which is introduced by omitting pixel-domain motion-compensation. Here, we use $D_{texture}$ to denote the macroblock distortion (SSD) between the original macroblock and the reconstructed macroblock with only texture quantization and without motion alteration. Let D$_{\text{motion}}$ denote the relative macroblock distortion (SSD) between the reconstructed macroblock without motion alteration and the reconstructed macroblock with motion alteration. We have observed that [8], the sum of the motion-introduced relative distortion and the texture-introduced distortion is approximated to the total resulted macroblock distortion (SSD) after texture quantization and motion alteration, that is,

$$D_{total} \approx D_{texture} + D_{motion}. \tag{3}$$

Moreover, in [9], it has been shown that D$_{\text{motion}}$ is highly independent of texture rate in a wild range. Based on (2), (3) and the independent relationship between texture-induced distortion and motion-induced distortion, the optimization problem in (1) can be formulated as follows:

$$\min J = \min(J_{motion}) + \min(J_{texture}), \tag{4}$$

Here,

$$\begin{aligned} J_{motion} &= D_{motion} + \lambda_{motion} \times R_{motion}, \\ J_{texture} &= D_{texture} + \lambda_{texture} \times R_{texture}. \end{aligned} \tag{5}$$

Thus, the rate-distortion optimization problem has been separated to two independent optimization problems: texture rate-distortion optimization and motion rate-distortion optimization.

3.2. Motion Rate-Distortion Optimization

In motion rate-distortion model, the motion rate can be achieved feasible during transcoding in DCT domain. And, it has been observed that [10], the relative distortion induced by motion alteration is highly related to the motion vector difference and the power spectral density of the video data, that is,

$$\begin{aligned} D_{motion} &\approx \varphi_x |\Delta mv_x|^2 + \varphi_y |\Delta mv_y|^2 + \varphi_{xy} |\Delta mv_x \Delta mv_y| \\ &\approx \varphi_x |\Delta mv_x|^2 + \varphi_y |\Delta mv_y|^2 \end{aligned} \tag{6}$$

Here, $(\Delta mv_x, \Delta mv_y)^t$ denotes the difference between the input motion and the output motion, and

$$\begin{aligned} \varphi_x &= \frac{1}{(2\pi)^2} \iint_{(-\pi,\pi]} S(\vec{\omega})\omega_1^2 d\omega_1 d\omega_2, \\ \varphi_y &= \frac{1}{(2\pi)^2} \iint_{(-\pi,\pi]} S(\vec{\omega})\omega_2^2 d\omega_1 d\omega_2, \\ \varphi_{xy} &= \frac{2}{(2\pi)^2} \iint_{(-\pi,\pi]} S(\vec{\omega})\omega_1\omega_2 d\omega_1 d\omega_2. \end{aligned} \tag{7}$$

Here, $\vec{\omega} = (\omega_1, \omega_2)^t$ denotes the two dimensional frequency, and $S(\vec{\omega})$ represents the power spectral density (PSD) of the prediction signal. In (6), the last approximation is made based on that the value of the third item is much smaller than the previous two ones. In fact, the prediction signal is not provided in DCT-domain transcoding. Instead, given φ_x and φ_y of each block in the reference frame and the input motion information, we can get φ_x and φ_y of current frame by weight-averaging the corresponding values of the motion-aligned blocks in the reference. The weight factors are determined with regard to the ratio of the overlapped pixels, as shown in Figure 4. Thus, only the PSD for P frames needs to be computed with 4x4 FFT transform or approximated with 4x4 DCT-like integer transform [5].

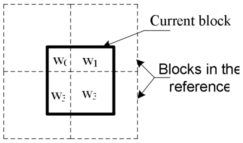

Figure 4. Computation of φ_x and φ_y by weight-averaging the corresponding values in the reference frame with motion aligned.

Because the B-pictures are also used as references, the distortion in the current frame would be propagated to the other frames. Considering bidirectional prediction and the pyramid structure as shown in Figure 1, the motion-induced distortion at decomposition level l is further given by [10]

$$D_{motion} \approx G_l(\varphi_x |\Delta mv_x|^2 + \varphi_y |\Delta mv_y|^2). \tag{8}$$

Here,

$$G_l = \frac{1}{4}[1 + 2\sum_{n=1}^{2^l}(1 - \frac{n}{2^l})^2]. \tag{9}$$

Because of independent relationship between motion-induced distortion and texture-induced distortion, equal slop is the optimal solution to allocate the bits between motion information and texture information, i.e.

$$\lambda_{motion} = \lambda_{texture}. \tag{10}$$

Further, the Lagrange multiplier in texture RD model has been evaluated in [8], that is,

$$\lambda_{texture} \approx \frac{1}{41} Q^{2.54}. \tag{11}$$

In practice, given a quantization parameter, the texture RD cost is unchangeable. Therefore, the mode and motion information can be selected when only minimizing

$$J_{motion} = D_{motion} + \lambda_{motion} \times R_{motion}. \tag{12}$$

Because we need to adjust the motion to adapt to lower bit-rate, the motion adjustment is achieved by a bottom-up mode integration process. For example, when the input mode is the 8x16 mode, four modes are considered as candidates: initial 8x16 mode, 16x16 mode with motion vectors from the left 8x16 block, 16x16 mode with motion vectors from the right 8x16 block and direct mode. The R-D cost is computed using (12) for each candidate and the minimal one is selected as the final macroblock mode. The texture information is directly re-quantized to form output stream.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Foreman, Carphone and Mobile are considered in our experiments. They are coded with single-layer H-B structures with QP 22 by [11] at CIF and fps 30Hz. The GOP size is set to 32. Then they are transcoded to FGS streams by different transcoders: *Motion_Reuse_Pixel* and *Motion_Reuse_DCT* denote the transcoders which are performed in pixel domain and in DCT domain, respectively. Both of them reuse the input motion information. *Our_Proposed* represents the transcoder which uses our proposed mode decision in DCT domain. All of transcoders generate two FGS layers besides base layers and perform P-frame transcoding in pixel domain. Our simulation platform is Win.XP running on P4 3.0GHz CPU and 1G RAM. The results are shown in Figure 5 and Figure 6.

It can be seen that, the DCT-domain open-loop transcoder achieves the same performance as that in pixel domain but speeds up the transcoding process significantly. And further, our proposed mode decision method in DCT domain can improve the coding efficiency up to 1dB at low bit-rate in terms of PSNR, and have loss only 0.5 dB at worst, and the proposed mode decision has little computational burden. The performance at high bit-rate suffers from loss because the motion has been adjusted for lower bit-rate.

In [12], adaptive motion refinement for FGS layers is proposed. If we transcode to FGS streams with motion refinement, the rest of the motion information, which is dropped by our mode decision in DCT domain can be re-encoded in FGS layers. Thus, the performance at high bit-rate will never suffer from loss.

5. CONCLUSIONS

In this paper, we consider the case that, once the scalable video is deployed extensively in multimedia systems, it is a need to transcode to scalable video streams from existing non-scalable coded streams. Specially, here, we investigate on transcoding to FGS streams from H.264/AVC hierarchical B-pictures. First, we design the fast transcoder architecture in DCT domain. Then we propose a mode decision method to select an RD optimal mode in DCT domain. Experimental results demonstrate that our proposed transcoder can achieve a good trade-off between the performances at low bit-rate and high bit-rate.

6. REFERENCES

[1] A. Vetro, C. Christopoulos, and H. Sun, “Video Transcoding Architectures and Techniques: an Overview”, IEEE *Signal Processing Magazine*, March 2003.

[2] J. Xin, C. –W. Lin, and M. –T. Sun, “Digital Video Transcoding”, *Proceedings of the* IEEE, Vol. 93, No. 1, Jan 2005

[3] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard”, IEEE *Trans. on CSVT*, Vol. 11, No. 3, Mar. 2001.

[4] ITU-T and ISO/IEC JTC1, “Scalable Video Coding – Joint Draft 4”, JVT-Q201, Oct. 2005.

[5] ITU-T and ISO/IEC JTC1, “Advanced Video Coding for Generic Audiovisual Services”, ITU-T Recommendation H.264 – ISO/IEC 14496 AVC, 2003

[6] H. Schwarz, D. Marpe and T. Wiegand, “Comparison of MCTF and Closed Loop Hierarchical B Pictures”, Joint Video Team, Doc. JVT-P059, Poznan, Poland, July 2005.

[7] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, “Rate-Constrained Coder Control and Comparison of Video Coding Standards”, IEEE *Trans. on CSVT*, Vol. 13, No. 17, July 2003.

[8] H. Shen, X. Sun, F. Wu, and S. Li, “Rate-Distortion Optimization for Fast Hierarchical B-Picture Transcoding”, IEEE International Symposium on Circuits and Systems, May, 2006, Island of Kos, Greece.

[9] R. Xiong, J. Xu, F. Wu and Y.-Q. Zhang, “Layered Motion Estimation and Coding for Fully Scalable 3D Wavelet Video Coding”, International Conference on Image Processing, 2004.

[10] A. Secker and D. Taubman, “Highly Scalable Video Compression With Scalable Motion Coding”, IEEE *Trans. on Image Processing*, Vol. 13, No. 8, August 2004.

[11] ITU-T and ISO/IEC JTC1, “JSVM 4 Software”, JVT-Q203, Oct. 2005.

[12] ITU-T and ISO/IEC JTC1, “Adaptive Motion Refinement for FGS Slices”, JVT-Q031, Oct. 2005.

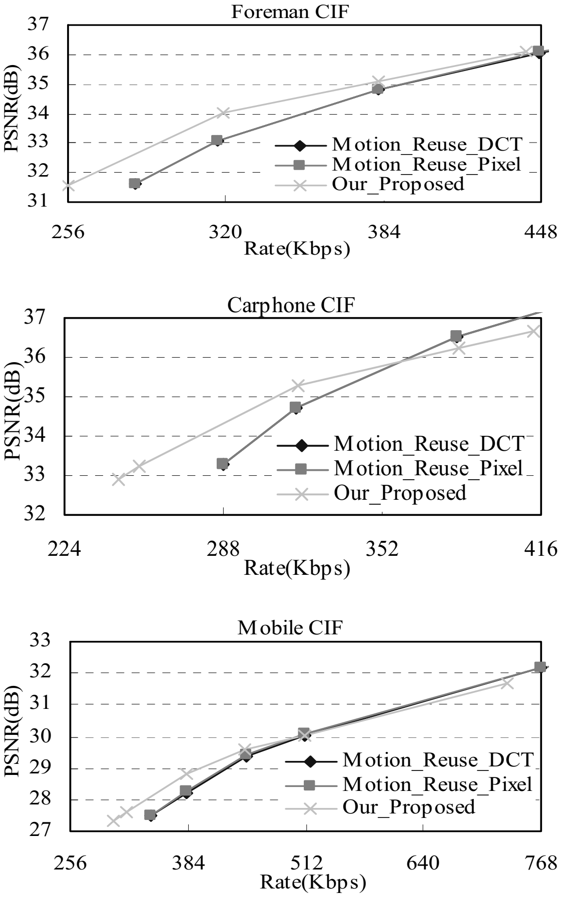

Figure 5. The coding efficiency comparison between different transcoders.

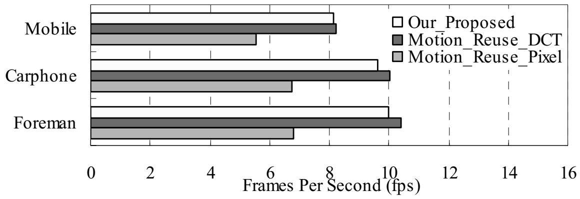

Figure 6. Frame rate comparison between different transcoders.