Fast H.264/MPEG-4 AVC Transcoding Using Power-Spectrum Based Rate-Distortion Optimization

Huifeng Shen, Xiaoyan Sun, and Feng Wu, *Senior Member, IEEE*

***Abstract*—Since variable block-size motion compensation (MC) and rate-distortion optimization (RDO) techniques are adopted in H.264/MPEG-4 AVC, modes and motion vectors (MVs) in input stream can no longer be reused equivalently efficient over a wide range of bit rate in transcoded streams. This paper proposes a new RDO model to maintain good coding efficiency and greatly reduce computation of the H.264/MPEG-4 AVC transcoding, in which the distortion caused by motion and mode changes is not calculated directly from the sum of absolute difference (SAD) or the sum of square difference (SSD) between source signals and interpolated prediction signals. Instead, distortion is directly estimated from MV variation and the power spectrum (PS) of the prediction signal generated from input stream. The proposed RDO model can be applied to both the pixel-domain transcoding and the transform-domain transcoding even when coded signals are not reconstructed at all. Furthermore, the techniques as to derive the Lagrangian multiplier in the proposed model are developed in respective pixel- and transform-domains. Additionally, we propose an H.264/MPEG-4 transcoding scheme that demonstrates the advantage of the proposed RDO model in terms of peak signal-to-noise ratio and transcoding speed, in which P-pictures are transcoded in the pixel domain for achieving reconstructed high quality and B-pictures are transcoded in the transform domain for high-transcoding speed.**

***Index Terms*—H.264/MPEG-4 AVC, picture power spectrum (PS), rate-distortion optimization (RDO), video coding, video transcoding.**

I. Introduction

NOWADAYS, more and more consumer electronic devices that have the capability of video playback, such as laptops, PDAs and even smart phones, are extensively used in media applications. However, these devices may be quite different in display resolutions, memory, processing powers, access bandwidths, and so on. How to make media contents especially video be free enjoyed among various devices becomes very challenging. Video adaptation through transcoding [1]–[3] is one of the most promising methods, which can provide bit-rate reduction, resolution reduction and format conversion to meet various requirements from devices as well as heterogeneous wired and wireless networks.

Manuscript received October 30, 2006; revised March 20, 2007 and March 22, 2007. This work was performed at Microsoft Research Asia, Beijing, China. This paper was recommended by Associate Editor H. Sun.

H. Shen is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: shenhf@mail.ustc.edu.cn).

X. Sun and F. Wu are with Microsoft Research Asia, Beijing 100080, China (e-mail: xysun@microsoft.com; fengwu@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.918783

A video transcoder can change coding parameters of a compressed video stream to generate another. In the early work on transcoding, researchers focused on bit-rate reduction to adapt to different access bandwidths. As more and more mobile devices with constrained display resolutions have been emerging in modern life, transcoding compressed video streams from a display resolution to a deducted one is extensively studied as well. Furthermore, since several video coding standards, such as H.261/3/4 and MPEG 1/2/4, exist for different video applications, transcoding video from a compressed format to another is also a need for the interoperability of video contents. All of the above cases can be implemented by directly concatenating a decoder and a corresponding encoder, where input stream is fully decoded and then the decoded pictures are re-encoded to target stream. This category of transcoders is the so-called pixel-domain transcoder because motion compensation (MC) is always performed in the pixel domain. It is drift-free with imperceptible degradation on visual quality compared with the case of directly coding raw video. But usually the transcoding in the pixel domain needs intensive computation. By exploiting the structural redundancy of the pixel-domain transcoder and the linear property of discrete cosine transform (DCT)/ inverse DCT (IDCT)/MC, the transform-domain transcoder is developed, in which decoded video is not completely reconstructed and transcoding is performed in transform domain. The transform-domain transcoder reduces computation but in return it would lead to quality degradation due to the drifting errors. So how to achieve the best tradeoff between computation and reconstructed video quality is the main focus in the current transcoding researches.

In this paper, we focus our attention to the H.264/MPEG-4 AVC (H.264 in short) transcoding from a higher bit rate to a lower bit rate. Format conversion from this standard to other standards is out of this paper's scope. The up-to-date H.264 video coding standard [4]–[6] achieves a significant improvement on coding efficiency by introducing some new techniques. However, most of them greatly influence the transcoding of the H.264 streams in terms of speed and quality. For example, intra prediction would propagate drifting errors to the whole image in the transform-domain transcoding; rounding and clipping operations in the half-/quarter-pixel interpolation make motion-compensation nonlinear; moreover, in-loop deblocking filters at the encoder and the decoder are nonlinear operations again. All of them make the cascaded loops of decoding and re-encoding impossible to be merged together mathematically equivalent. In other words, these techniques would prevent fast transcoding of successive pictures in transform domain because they would cause severe drifting errors [3]. But the transform-domain transcoder may be still feasible when there are only short-term/no dependencies among frames.

In pixel-domain transcoders, the motion-vector (MV)-reuse approach has been demonstrated to be an efficient way to achieve a good tradeoff between computation and quality in transcoding of streams generated by previous video coding standards (e.g. MPEG-2) [3]. However, since H.264 employs variable block-size MC and rate-distortion optimized (RDO) motion and mode selection [6], motion and mode data used at high bit rates is quite different from that at low bit rates. Therefore, directly using input motion and mode data during transcoding would cause significant degradation on coding efficiency. Unfortunately, since the conventional RDO motion and mode selection needs intensive computation, the re-encoding process would prevent the pixel-domain transcoder from real-time applications especially for high-resolution video. Moreover, the conventional R-D model cannot be applied to transform-domain transcoding because there are no reconstructed pictures available.

Several solutions have been reported in the literature to address the above problems in pixel-domain transcoding. Zhang *et al.* [7] propose a mode mapping for the H.264 spatial resolution reduction, which is time-saving but suffers from more than 3.0 dB losses. In [8] and [9], fast motion estimation and mode refinement have been proposed by limiting possible modes as well as searching points for bit-rate reduction. [10] proposes the area-weighted vector median motion estimation, and [11] presents an approach that includes bottom-up motion re-estimation, fast mode selection and adaptive motion refinement for the H.264 resolution reduction. The techniques in [8]–[11] can achieve a comparable coding efficiency with the fully re-encoding. However, since for a macroblock multiple modes have to be evaluated in the motion re-estimation or motion refinement and a mass of sum of absolute difference/sum of square difference (SAD/SSD) computation as well as interpolation has to be involved, these techniques still need intensive computation. In addition, to the authors' best knowledge, little work has been involved in the RDO problem in transform-domain transcoding so far.

Therefore, besides reducing possible modes and motion search points, the more important is whether we can avoid fractional-pixel interpolation and SAD/SSD calculation to speed up the H.264 transcoding. Furthermore, to enable RDO in transform-domain transcoding, can the R-D cost be estimated without raw signal? Motivated by Secker's work on scalable MV coding for the wavelet-based video coding [15], we propose a new RDO model for the H.264 transcoding. In our proposed model, the distortion, which is defined as the SSD of two predictions generated from two sets of MV, is estimated based on the differences of two sets of MV and the power spectrum (PS) of the prediction generated from input MVs. Since the computation of the prediction's PS still needs interpolation, we further propose techniques to approximate the PS of prediction in pixel-domain and transform-domain, respectively. Thus, our distortion computation is free from SAD/SSD computation, subpixel interpolation and also original signal.

In our proposed R-D model, the Lagrangian multiplier is different in pixel-domain transcoding and in transform-domain transcoding. In pixel-domain transcoding, we borrow the technique used in H.264. However, in transform-domain transcoding, different MVs used in the encoder and decoder contribute distortion in addition to quantization. According to [17], the distortions caused by quantization and motion mismatch are independent. Consequently, the R-D cost of our proposed model can be split into two parts: one related to quantization and the other to motion adjustment. Following the equal slope principle, the Lagrangian multipliers in quantization and motion parts are derived from the relationship of quantized distortion and used residual bits. Furthermore, the derived Lagrangian multipliers are adjusted by taking error propagation into account in the case of hierarchical B-picture transcoding. Our proposed RDO model and its applications in different transcoders have been first reported in our conference papers [12]–[14].

The rest of this paper is organized as follows. Section II discusses the proposed R-D model and the distortion estimation based on the PS function. The techniques to calculate Lagrangian multipliers in our R-D model are presented in Section III. The proposed H.264 transcoding scheme for bit reduction is described in Section IV. Section V presents the experimental results of various transcoding scenarios. Section VI concludes this paper.

II. Proposed RDO

In this section, we will first review the conventional RDO method used in the H.264 encoder and point out the problems when it is applied to its transcoding. Then our proposed RDO model for the H.264 transcoding is discussed in detail.

A. Conventional Rate-Distortion Optimization

H.264 employs multiple macroblock partition modes for motion-compensated prediction. The Lagrangian optimization method is used to find the optimum MV for inter-coded block and optimum coding mode for each macroblock. It can achieve optimum bit allocation between motion data and residual data. In the H.264 RDO method, the rate-constrained motion estimation is first performed to find optimum MV for an inter-coded block by minimizing

$$J(B, I|\lambda_{\text{motion}}) = D_{\text{dfd}}(B, I) + \lambda_{\text{motion}} R_{\text{motion}}(B, I). \quad (1)$$

Here, B denotes an encoding block and I represents coding parameters for B. Distortion $D_{\text{dfd}}(B, I)$ denotes the difference between original block and prediction of B with parameters I, which is represented by SAD or SSD. R_{motion} is the number of bits to code MVs of this block. The Lagrangian multiplier λ_{motion} is adjusted dependent on the use of SAD or SSD. If the distortion is represented by SSD, the Lagrangian multiplier λ_{motion} is equal to

$$\lambda = 0.85 \times 2^{(\text{QP}-12)/3} \quad (2)$$

where, QP denotes quantization parameter (QP) used in H.264. Otherwise, the Lagrangian multiplier λ_{motion} is equal to $\sqrt{\lambda}$.

After MVs are selected, rate-constrained mode selection is employed to find optimum coding mode by minimizing

$$J(B, I|\lambda_{\text{mode}}) = D_{\text{rec}}(B, I) + \lambda_{\text{mode}} R_{\text{total}}(B, I). \quad (3)$$

Here, D_{rec} denotes the SSD between an original block and its reconstruction. R_{total} is the number of bits for coding quantized

transform residual data as well as motion data. The Lagrangian multiplier λ_{mode} in (3) is equal to the λ in (2).

Notice that the prediction in (1) is usually generated by half-/quarter-pixel interpolation because of fractional MV. Furthermore, the distortion has to be calculated for each search point in motion estimation, which involves a mass of SAD/SSD computation. Thus, this optimization process is of high computational intensity. In (3), one macroblock has to be coded repeatedly to obtain its reconstructed distortion and used bits for every candidate mode. The computation of this process is even higher. Therefore, the existing RDO method is hard to be applied to real-time transcoding. In addition, the optimization in (1) and (3) cannot be directly used in transform-domain transcoding due to the lack of reconstructed pictures.

B. Proposed Rate-Distortion Model

To further reduce the complexity of RDO motion estimation and mode selection, a new R-D model is proposed for the fast H.264 transcoding, in which the distortion is estimated from MV variation and picture PS.

In rate-reduction transcoding, input stream is usually of high quality and high bit rate. Motion data used in input stream can be granted as a precise representation of real motion and needs many bits to code. However, these motion bits would be a heavy burden in low bit rate coding so as to hurt coding performance. Without any adjustment on motion data, the transcoder cannot operate equivalently efficient over a wide range of bit rate. Assuming that input MVs and modes are an optimum point at the input bit rate, the RDO problem in transcoding can be generalized as reshaping input modes as well as MVs to another optimum point. It motivates us to think about the RDO problem by taking the input modes and MVs as a start point and find the optimum MVs and modes at a new bit rate.

For convenience, we use mv_h to represent input MV for the current block hereafter. p_h is the prediction block generated using mv_h and the previous transcoded frame. Let MV_L represent the set of candidate MVs in the transcoded stream. p_i is the prediction block resulting from $\mathrm{mv}_i(\mathrm{mv}_i \in \mathrm{MV}_L)$ and the previous transcoded frame. To select optimum MVs and mode in transcoding, we propose an RDO model with the similar form of (1) and (3), where MVs and mode are selected by minimizing

$$J(B, I|\lambda_p) = D_p(B, I) + \lambda_p R_p(B, I). \tag{4}$$

Here R_p is defined as

$$R_p(B, I) = R_h(B) - R_i(B, I). \tag{5}$$

Here $R_h(B)$ represents the number of bits to code input MV and mode of block B, whereas $R_i(B, I)$ denotes the number of bits to code the candidate MV and mode of block B with parameter set I. The distortion D_p represents the SSD between two prediction blocks resulting from two different sets of MV but with the same reference, that is

$$D_p(B, I) = \sum_{n \in B} (p_i[n] - p_h[n])^2 \tag{6}$$

where n is the pixel index in block B. To reduce the computation of RDO, we would not like to calculate p_i and D_p for each candidate MV mv_i. Instead, if p_i is modeled as a function of p_h, the distortion D_p can be directly calculated free from SSD computation. Similar to the scalable MV coding work [15] for wavelet-based video coding, the distortion is estimated by power spectral density (PSD) of prediction p_h and the difference between two sets of MV mv_h and mv_i. The brief derivation is given here for the self-contained purpose.

For convenience, we use $P_h(\omega_1, \omega_2)$ to denote the prediction p_h after Fourier transform. Here, $\omega = (\omega_1, \omega_2)$ is a 2-D frequency row vector. If an input MV is adjusted to a new one, their difference is represented as $\Delta\mathrm{mv} = (\Delta\mathrm{mv}_x, \Delta\mathrm{mv}_y)^t$. The prediction block from the new MV is p_i, which is also denoted by $P_i(\omega_1, \omega_2)$ in frequency domain. Since pixel shift corresponds to linear phase shift in frequency domain after Fourier transform, we have $P_i(\omega_1, \omega_2) = e^{-j\omega\Delta\mathrm{mv}} P_h(\omega_1, \omega_2)$. According to the Parseval's theorem, the sum of square errors, as denoted by D_p between p_h and p_i, can be calculated in frequency domain by

$$D_p = \frac{1}{(2\pi)^2} \int\int_{(-\pi,\pi)} S_h(\omega_1, \omega_2) \cdot (1 - e^{-j\omega\Delta\mathrm{mv}})^2 d\omega_1 d\omega_2. \tag{7}$$

$S_h(\omega_1, \omega_2)$ is the power spectral density of p_h. The Taylor expansion of $(1 - e^{-j\omega\Delta\mathrm{mv}})^2$ yields

$$(1 - e^{-j\omega\Delta\mathrm{mv}})^2 = \frac{2(\omega\Delta\mathrm{mv})^2}{2!} - \frac{2(\omega\Delta\mathrm{mv})^4}{4!} + \frac{6(\omega\Delta\mathrm{mv})^6}{6!} - \cdots \tag{8}$$

For small values of $(\omega\Delta\mathrm{mv})$, by omitting high-order terms in (8), we have the approximation as follows:

$$D_p = \frac{1}{(2\pi)^2} \int\int_{(-\pi,\pi]} S_h(\omega_1, \omega_2) \cdot (\omega\Delta\mathrm{mv})^2 d\omega_1 d\omega_2. \tag{9}$$

Equation (9) can be further expressed as

$$\begin{aligned} D_p &\approx \varphi_x \Delta\mathrm{mv}_x^2 + \varphi_y \Delta\mathrm{mv}_y^2 + \varphi_{xy} |\Delta\mathrm{mv}_x \Delta\mathrm{mv}_y| \\ &\approx \varphi_x \Delta\mathrm{mv}_x^2 + \varphi_y \Delta\mathrm{mv}_y^2 \end{aligned} \tag{10}$$

where φ_x, φ_y, and φ_{xy} are

$$\begin{aligned} \varphi_x &= \frac{1}{(2\pi)^2} \iint_{(-\pi,\pi]} S_h(\omega_1, \omega_2) \cdot \omega_1^2 d\omega_1 d\omega_2 \\ \varphi_y &= \frac{1}{(2\pi)^2} \iint_{(-\pi,\pi]} S_h(\omega_1, \omega_2) \cdot \omega_2^2 d\omega_1 d\omega_2 \\ \varphi_{xy} &= \frac{2}{(2\pi)^2} \iint_{(-\pi,\pi]} S_h(\omega_1, \omega_2) \cdot \omega_1 \omega_2 d\omega_1 d\omega_2. \end{aligned} \tag{11}$$

The final approximation in (10) is made based on the fact that the value of $\varphi_{xy}|\Delta\mathrm{mv}_x \Delta\mathrm{mv}_y|$ is usually much smaller than that of $\varphi_x \Delta\mathrm{mv}_x^2 + \varphi_y \Delta\mathrm{mv}_y^2$.

The power spectral density of p_h is conventionally estimated by the square of 2-D fast Fourier transform (FFT) of each block. Considering the proposed transcoding scheme is built upon H.264, a more simple method is to use integer 4×4 DCT-like transform [16] instead of FFT, where the coefficients after 4×4 integer transform are properly scaled in the same way as the case that QP is 4 [16]. These coefficients represent the estimated spectrum magnitudes at discrete frequency points

$\pm 2\pi u/2N$ with $u = 0, 1, 2, 3$ and $N = 4$. Modifying (11) into discrete forms, φ_x and φ_y are easy to get by setting ω_1 and ω_2 as $\pm 2\pi u/2N$.

Let's qualitatively analyze our proposed model and the conventional one on calculating distortion in terms of computation. In the R-D models as shown in (1) and (3), the distortions for each block and for each check point are obtained through SSD/SAD computation and subpixel interpolation if MV points to a half-/quarter-pixel position. Moreover, the transcoder has to access the reference picture in memory according to the changing of checking point during motion estimation. It greatly increases data traffic in transcoding. Furthermore, the conventional R-D model cannot be directly applied to transform-domain transcoding because there are no reconstructed signals available.

In our proposed model, the distortion between two predictions is estimated by (10) in frequency domain, which only needs to calculate MV difference for each checking point. In the worst case, we need to calculate one differential vector (i.e., $\Delta\mathrm{mv}$) for each 4×4 block corresponding to the 4×4 block mode; whereas in the best case, only one $\Delta\mathrm{mv}$ for each 16×16 block is computed corresponding to the 16×16 block mode. Moreover, φ_x and φ_y can be calculated before the mode decision of this block, and are independent of the number of checking points. In the case of a mass of checking points, the computation burden of φ_x and φ_y is still constant and can be negligible. Thus, our proposed model is free from SAD/SSD computation, subpixel interpolation and vast memory access during RDO. In addition, (10) provides a way to estimate distortion without reconstruction so that it enables RDO in transform-domain transcoding.

C. Approximating φ_x and φ_y of Prediction

Equation (10) indicates that we need to generate prediction first from input motion data to calculate φ_x and φ_y. In this process, subpixel interpolation may still be performed on references if input MVs are of fractional precision, which is time consuming. In pixel-domain transcoding, current reconstructed pictures of input stream are always available as the source to generate transcoded stream. Considering input MV as a precise representation of source motion and strong correlation among neighboring temporal pictures, spatial textural feature of prediction signal is very similar to that of the current reconstructed signal especially at regular motion regions. Therefore, it is reasonable to approximate the PS of prediction block by that of input reconstructed block so that the interpolation for generating prediction is avoided. That is, φ_x and φ_y of prediction blocks are approximated by those of input reconstructed blocks. T

A much tough problem is how to approximate φ_x and φ_y of prediction in transform-domain transcoding because reconstructed blocks are not available there. Transform-domain transcoding schemes are widely used in B-pictures. Although H.264 allows B-pictures to be predicted from B-pictures in the hierarchical B-picture structure, the prediction path of B-pictures is usually relatively shorter than that of P-pictures. In this case, error accumulation is not as severe as that in P-pictures. In addition, since P-pictures are transcoded in pixel-domain in our proposed scheme, the reconstructed P-pictures severing as references in B-picture coding are always available. Therefore, in this paper we propose a technique to estimate φ_x and φ_y of prediction from the forward and backward references in transform-domain transcoding for B-pictures.

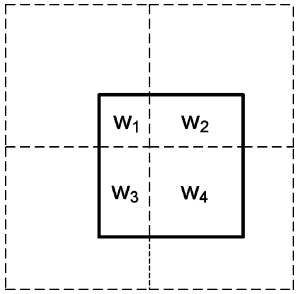

Fig. 1. Computation of φ_x and φ_y through weight-averaging the corresponding values in the motion aligned reference blocks.

In our transform-domain transcoder, φ_x and φ_y of reference blocks can be calculated by the conventional method. Since the block size is very small (e.g., 4×4 in this paper) and there is strong correlation among neighboring pixels, we assume that the spatial texture feature is uniform within such a small 2-D-block. Based on this assumption, the PS of the motion-compensated block can be approximated by weight-averaging the PS from four overlapped blocks. As the integral operator is linear, we can estimate φ_x and φ_y of prediction by weight-averaging the corresponding values of the motion aligned blocks in the reference, i.e.,

$$\begin{cases} \varphi_{x,t}[j] \approx \sum_i w_i \varphi_{x,t'}[i] \\ \varphi_{y,t}[j] \approx \sum_i w_i \varphi_{y,t'}[i]. \end{cases} \tag{12}$$

Here, t indicates the current picture and t' represents the forward or backward reference. j is the current block and i denotes the blocks in the reference covered partially or completely by current block. The weight factors w_i are determined with regard to the ratio of overlapped pixels, as shown in Fig. 1. The square block with solid borders denotes a 4×4 block in the current frame, and the square blocks with dashed borders denote the corresponding motion-aligned blocks in the reference frame. The weight factors w_1, w_2, w_3, and w_4 correspond to the ratios of the overlapped pixels. When the hierarchical B-picture structure is supported, φ_x and φ_y of the B-pictures can be estimated recursively using the approach (12).

III. Lagrangian Multiplier in Our Proposed RDO Model

In this section, we will discuss how to derive the Lagrangian multiplier in the proposed RDO model. Although the same RDO model is used in pixel-domain and transform-domain transcoding schemes, different techniques are used to derive Lagrangian multiplier because the reconstructed video distortion in pixel-domain transcoding is caused by quantization only but the reconstructed video distortion in transform-domain transcoding is caused by both quantization and MV mismatch.

A. Lagrangian Multiplier in Pixel Domain

How to derive Lagrangian multiplier in pixel-domain transcoding is a relatively simple issue because the similar problem has been addressed in H.264. In pixel-domain transcoding, an input stream is fully decoded and the reconstructed video is served as source. The proposed technique

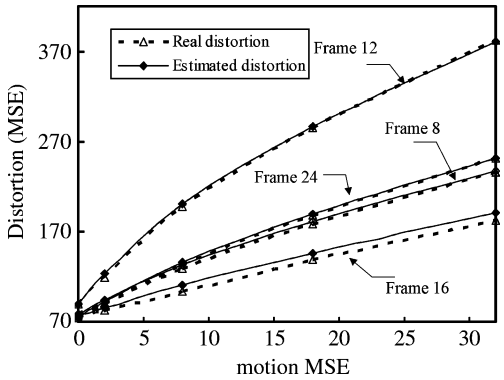

Fig. 2. Comparisons of D_{total} and $(D_{\text{motion}} + D_{\text{texture}})$ in terms of MSE.

simplifies the calculation of distortion but does not modify the relationship of distortion and used bits. In other words, Lagrangian multiplier should be the same as that in H.264. Since SSD is used in our proposed RDO model, λ_{pixel} in pixel-domain transcoding is

$$\lambda_{\text{pixel}} = \lambda. \tag{13}$$

B. Lagrangian Multiplier in Transform Domain

Generally, total coded rate consists of two parts: motion rate and texture rate, denoted as

$$R_{\text{total}} = R_{\text{motion}} + R_{\text{texture}}. \tag{14}$$

Here R_{motion} is the number of the bits spent in coding submacroblock/macroblock modes and MVs, while R_{texture} is the number of the bits in coding quantized transform coefficients. In traditional transform-domain transcoding schemes, rate reduction is achieved by only amplifying quantization, namely, reducing R_{texture}, whereas MVs and modes are simply reused. However, when the RDO method is applied to H.264, R_{motion} also needs to be downscaled; otherwise too many bits would be spent in coding motion data at low bit rates.

How to optimally reduce motion rate as well as texture rate is a new problem in the fast transform-domain transcoding. In this case, two kinds of distortions are introduced: one from re-quantized transform coefficients and the other from motion data which is modified to a coarse representation without a pixel-domain motion re-compensation loop. Let D_{texture} denote the former, where motion data is completely reused in transcoding. Let D_{motion} denote the latter, where re-quantization is not modified at all. Then let D_{total} denote the total distortion introduced by both texture quantization and motion adjustment during transform-domain transcoding.

Experiments have been designed to analyze the relationship among D_{total}, D_{motion}, and D_{texture} in the H.264 B-picture coding. Fig. 2 shows the typical relationship among them by taking the 8th, 12th, 16th, and 24th frames of Foreman (CIF) sequence as examples. These frames are coded as hierarchical B-frames in the GOP of 32 with a fixed QP, as depicted in Fig. 2. And during transcoding, these frames are re-quantized by another fixed QP and motion-adjusted by different MV variations. The Δmv is manually set to (1, 1), (2, 2), (3, 3), and (4, 4) at quarter-pixel resolution respectively and the motion-induced distortions (that is, D_{motion}) are increasing correspondingly. The actual values of D_{total} are depicted by the dashed lines in Fig. 2, whereas the solid lines represent the values of $(D_{\text{motion}} + D_{\text{texture}})$. One can observe that the total distortion D_{total} in B-picture transcoding can be well approximated by the sum of distortions, D_{motion} and D_{texture}, i.e.,

$$D_{\text{total}} \approx D_{\text{motion}} + D_{\text{texture}}. \tag{15}$$

It has been reported in [17] that D_{motion} is highly independent of texture-induced distortion. So, according to (14) and (15), the RDO model can be separated to texture RDO model and motion RDO model as:

$$\begin{aligned}\min_I J(B,I|\lambda) = {} & \min_I J_{\text{motion}}(B,I|\lambda_{\text{transform}}) \\ & + \min_I J_{\text{texture}}(B,I|\lambda_{\text{transform}})\end{aligned} \tag{16}$$

where,

$$\begin{aligned}& J_{\text{motion}}(B,I|\lambda_{\text{transform}}) \\ & \quad = D_{\text{motion}}(B,I) + \lambda_{\text{transform}} R_{\text{motion}}(B,I) \\ & J_{\text{texture}}(B,I|\lambda_{\text{transform}}) \\ & \quad = D_{\text{texture}}(B,I) + \lambda_{\text{transform}} R_{\text{texture}}(B,I).\end{aligned} \tag{17}$$

The motion R-D optimization can be achieved by modification of MVs and submacroblock/macroblock modes, while the texture R-D optimization can be achieved by adjusting QPs. In the optimization (16), Lagrangian multipliers in two terms should be equal to each other according to the equal slope principle. Therefore, we use $\lambda_{\text{transform}}$ to denote them.

C. Lagrangian Multiplier of One Picture

We will derive $\lambda_{\text{transform}}$ from the texture quantization optimization. As the texture R-D optimization is separated from the motion R-D optimization in our proposed transform-domain R-D model, it is inferable that $J_{\text{texture}}(B,I|\lambda_{\text{transform}})$ is only determined by QP as well as Lagrangian multiplier and is irrelevant to macroblock mode and MVs.

If the distortion-rate function $D_{\text{texture}}(R_{\text{texture}})$ is strictly convex, the minimum of the Lagrangian cost function is given by setting its derivative to zero [6], i.e.,

$$\frac{\partial J_{\text{texture}}}{\partial R_{\text{texture}}} = \frac{\partial D_{\text{texture}}}{\partial R_{\text{texture}}} + \lambda_{\text{transform}} = 0 \tag{18}$$

which yields

$$\lambda_{\text{transform}} = \frac{\partial D_{\text{texture}}}{\partial R_{\text{texture}}}. \tag{19}$$

In the derivation of Lagrangian multiplier, the models of rate and distortion corresponding to QP presented in [18] are used, i.e.,

$$\begin{cases} R_{\text{texture}} \approx aQ^{-\alpha} \\ D_{\text{texture}} \approx bQ^{\beta} \end{cases} \tag{20}$$

which assume that the transform coefficients have a Cauchy distribution and a uniform quantizer is used with step size Q. a, b, α and β are parameters which depend on video content.

Then we get

$$\left|\frac{\partial D_{\text{texture}}}{\partial R_{\text{texture}}}\right| = \left|\frac{\partial D_{\text{texture}}}{\partial Q} \times \frac{\partial Q}{\partial R_{\text{texture}}}\right| \approx cQ^{\gamma}. \tag{21}$$

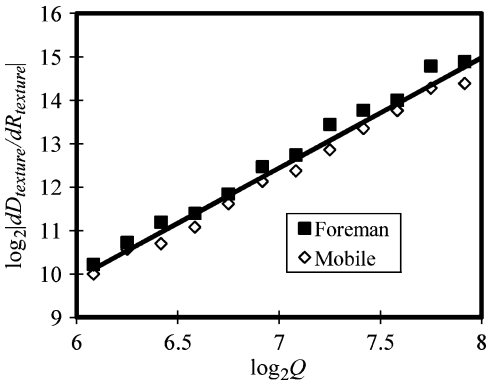

Fig. 3. Relationship between $\partial D_{\text{texture}}/\partial R_{\text{texture}}$ and Q.

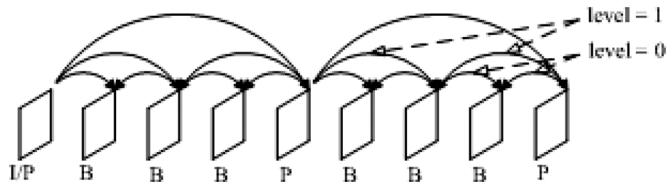

Fig. 4. Typical hierarchical B-coding structure (GOP size is 4). The solid lines with arrows mean the prediction directions.

Here, $c = b\beta/a(\alpha+1)$ and $\gamma = \alpha+\beta$. (21) can also be derived to a linear expression, that is

$$\log_2\left|\frac{\partial D_{\text{texture}}}{\partial R_{\text{texture}}}\right| \approx \gamma \log_2 Q + \log_2 c. \tag{22}$$

To ascertain parameters γ and c, some compressed streams are transcoded in transform-domain open-loop structure to different low bit rates with different Qs by reusing the input motion data. The results of Foreman and Mobile sequences are showed in Fig. 3. The bold line is the linear approximation of real data with the least square method. It can be described as

$$\log_2\left|\frac{\partial D_{\text{texture}}}{\partial R_{\text{texture}}}\right| \approx 2.54 \log_2 Q - 5.35. \tag{23}$$

So the approximation of the relationship between the quantizer Q and the Lagrangian multiplier can be described as

$$\lambda_{\text{transform}} = \left|\frac{\partial D_{\text{texture}}}{\partial R_{\text{texture}}}\right| \approx \frac{1}{41} Q^{2.54}. \tag{24}$$

D. Lagrangian Multiplier Considering Error Propagation

In H.264, B-pictures can be predicted from B-pictures. For example, Fig. 4 depicts a hierarchical B-coding structure, where B-pictures are organized into several levels. B-pictures at lower level are predicted from those at higher level. In such a structure, errors of B-pictures at higher level will be propagated into those B-pictures at lower level when reconstructing these pictures. Considering this error propagation, the estimated distortion should be multiplied by different energy gain factors at different temporal levels. Equivalently, the Lagrangian multiplier is divided by the energy factor so that the RDO model still has the same form as (4). For the case depicted in Fig. 4, the Lagrangian multiplier at level l should be

$$\lambda_{\text{transform}} \approx \frac{1}{41} Q^{2.54}/G_l. \tag{25}$$

G_l denotes the energy gain factor considering distortion propagation at the temporal level l. According to [15], it can be formulated as

$$G_l = 1 + 2\sum_{n=1}^{2^l}\left(1 - \frac{n}{2^l}\right)^2. \tag{26}$$

Here we assume that most of blocks are bi-directionally predicted in the input high bit rate stream. According to (10) and (25), the motion-induced R-D cost is able to be estimated without pixel-domain reconstruction in transform-domain transcoding.

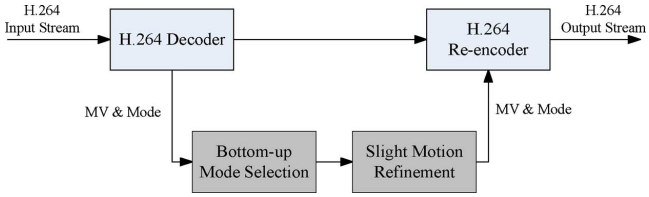

Fig. 5. Block diagram of the proposed transcoding scheme in pixel domain.

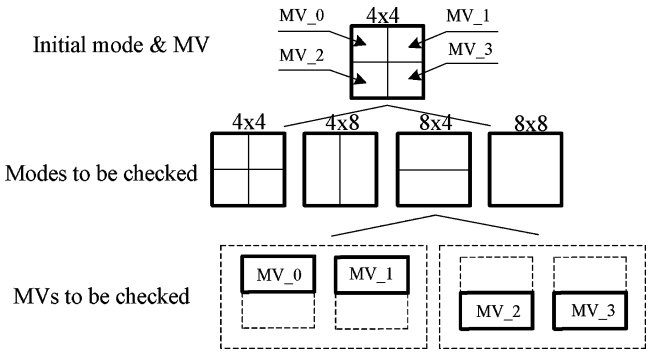

Fig. 6. Bottom-up submacroblock mode selection.

IV. Proposed Transcoding Schemes

This section presents the proposed transcoding schemes in pixel domain and transform domain to demonstrate the advantages of our proposed RDO model.

A. Transcoding Scheme in Pixel Domain

Fig. 5 illustrates the block diagram of our proposed transcoding scheme in pixel domain. The input H.264 stream is fully decoded first. The decoded video is regarded as original video for the H.264 re-encoder, in which motion estimation and mode decision do not exist. The extracted MVs and modes are used to aid the mode selection and motion refinement processes.

In general, low bit rate stream prefers large block-size modes to small block-size modes. Thus, in our scheme, the mode selection process is accelerated by the bottom-up mode constraint. It starts from the initial input MV and mode which are the entropy-decoded ones. For example, if the initial macroblock mode is 8

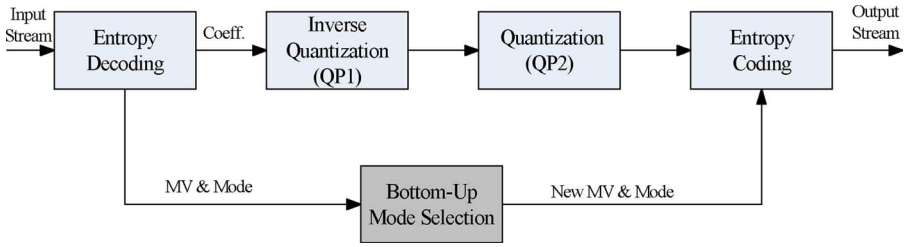

Fig. 7. Block diagram of the transcoding scheme in transform domain.

× 8 mode (including 8 × 8, 8 × 4, 4 × 8 and 4 × 4 submacroblock modes), the bottom-up submacroblock mode selection is first performed before the bottom-up macroblock mode selection. The submacroblock mode selection routes from the initial mode to larger block-size modes. If the initial mode of current submacroblock is 4 × 4, submacroblock modes 4 × 4, 4 × 8, 8 × 4 and 8 × 8 will be checked, as illustrated in Fig. 6. The mode that has the minimum cost according to (4) is selected as the final submacroblock mode. Along with the mode selection, the motion selection is also performed subject to (4). As depicted in Fig. 6, when the cost of the 8 × 4 submacroblock mode is checked, the MVs of the two corresponding 4 × 4 blocks will be used in the cost calculation, respectively. The MV that has less cost is selected for this partition. After submacroblock mode is decided, the rest of modes, 8 × 16, 16 × 8 and 16 × 16, will be checked by the mode selection with the same principle to finally determine the partition mode of this macroblock.

Motion refinement is introduced after the mode selection to maintain good coding efficiency for the pixel-domain P-frame transcoding. In our scheme, the resulting MV obtained by mode selection is used as the search center of the motion refinement. The search range of the motion refinement can be variable to pursuit different coding performance. In our scheme, as the initial MVs are obtained by the proposed R-D mode selection, the initial MVs are usually reliable so that the search range is only 1 integer pixel which involves eight integer-pixel positions, eight half-pixel positions and eight quarter-pixel positions. The MVs of the macroblock are determined by minimizing the cost defined in (1).

B. Transcoding Scheme in Transform Domain

Fig. 7 illustrates the block diagram of our transform-domain transcoding scheme. The input stream is only entropy-decoded first. The extracted transform coefficients are de-quantized with the input QP (QP1). Then the de-quantized coefficients are directly re-quantized with the new QP (QP2), and the re-quantized coefficients are entropy-encoded to generate the output stream. In addition, the new MVs and modes are generated after the proposed bottom-up mode decision which has been discussed in the above subsection.

V. Experimental Results

Experiments are designed to evaluate the performances of our proposed RDO model as well as the transcoding schemes in both pixel domain and transform domain. First, the accuracy on calculating φ_x and φ_y using 4 × 4 DCT-like integer transform is verified. Second, our proposed P-picture transcoding scheme is used for bit-rate reduction in pixel domain. Finally, hierarchical

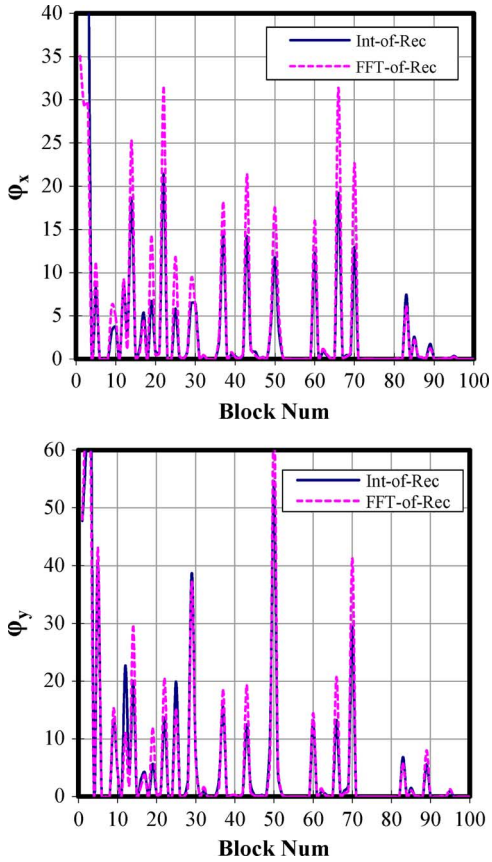

Fig. 8. Computing φ_x and φ_y of reconstructed blocks by 4 × 4 FFT/integer transform.

B-pictures are transcoded in transform domain to demonstrate the performance of our proposed method. We select JSVM1 [19] as benchmark to implement our schemes although both JSVM1 and JM [20] support hierarchical B-coding. Our simulation platform is Windows XP running on P4 3.0 GHz with 1 GB RAM.

A. Approximating φ_x and φ_y

In general, the PS is estimated by the square of FFT. In our scheme, 4 × 4 DCT-like integer transform replaces FFT to calculate φ_x and φ_y of the input reconstructed block. Therefore, this experiment is used to compare φ_x and φ_y by 4 × 4 DCT with those by 4 × 4 FFT.

In this experiment, the first two frames of "Foreman" sequence at CIF resolution are coded as I frame and P-frame, respectively, with QP = 22. The φ_x and φ_y of the first 100 reconstructed 4 × 4-blocks in the second frame are shown in

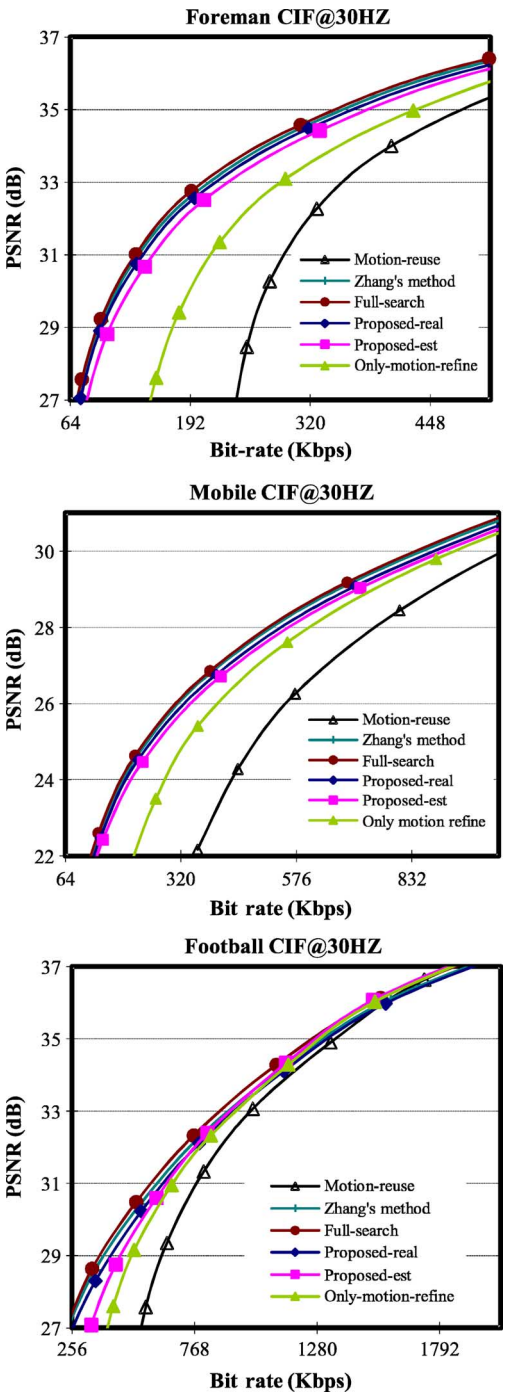

Fig. 9. Coding efficiency comparisons among different transcoders for bit-rate reduction at CIF resolution.

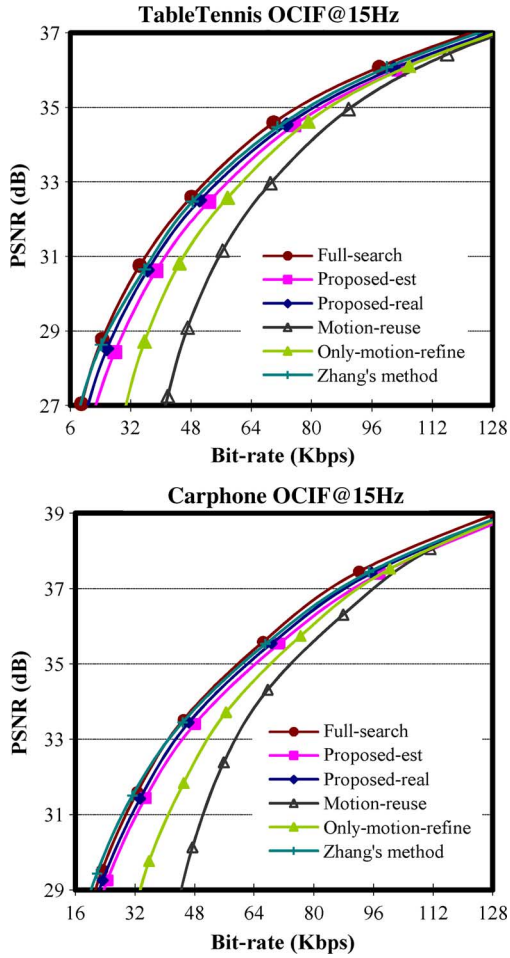

Fig. 10. Coding efficiency comparisons among different transcoders for bit-rate reduction at QCIF resolution.

Fig. 8. From the curves, one can observe that φ_x and φ_y of reconstructed blocks can be well approximated by those calculated by 4×4 integer transform which is computationally more simple than FFT.

B. P-Frame Transcoding in Pixel Domain

In this experiment, the sequences ("Foreman," "Mobile," and "Football" at CIF resolution with 30 fps, "Carphone" and "Table Tennis" at QCIF resolution with 15 fps) are coded by the H.264 main profile with RDO enabled. QP is fixed at 22. FREXt is disabled. The first frame is coded as I-frame, others are all P-frames and only one reference frame is used.

Several different transcoders are implemented in this experiment for comparisons: In the *Full-search* transcoder, the high bit rate stream is fully decoded first and then the decoded sequence is re-encoded by the H.264 encoder with the full-scale motion search whose range is 16; in the *Motion-reuse* transcoder, the MVs and modes are directly re-used; the *Proposed-est* transcoder means that the R-D optimization follows the criteria defined in (4), in which the distortion is estimated and free from SSD computation during motion selection; the *Proposed-real* transcoder is the same as the *Proposed-est* one, except that the prediction errors used in the bottom-up mode selection is obtained by computing SSD; the *Only-motion-refine* transcoder means that the transcoder re-uses input MVs and modes and refines the MVs within $+/-1$ search window following the criteria defined in (1), which is commonly used for previous coding standards; the *Zhang's method* proposed in [8] is also considered for comparison in this paper.

Figs. 9 and 10 show the coding efficiency comparisons among different transcoders for bit-rate reduction, while Fig. 11 illustrates the transcoding speed comparisons in terms of processed frames per second for "Foreman" and "Carphone". One can observe that our proposed transcoding schemes can achieve com-

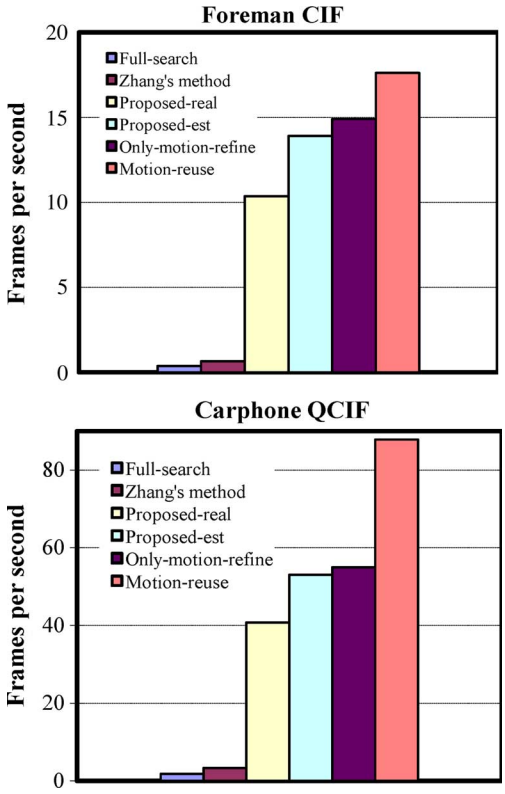

Fig. 11. Complexity comparisons in terms of frames per second among different transcoders.

parable coding performance in terms of PSNR. At the very low bit rate on Football, which is the worst case, the loss between our proposed and the full-search one is about 1.0 dB. However, Fig. 11 indicates that our proposed method can accelerate the transcoding process more than ten times compared with the Zhang's method.

C. Hierarchical B-Picture Transcoding in Transform Domain

Three test sequences, "Foreman," "Mobile," and "Football" at CIF resolution, are used in this test. The source streams to be transcoded are generated by JSVM1 [19] with the hierarchical B-structure. The GOP size is 32. The main profile is used and FREXt is disabled. The first frame is encoded as I frames, one P-frame is inserted every 32 frames, and other frames are hierarchically coded as B-frames. The constrained intra mode is used. It means intra blocks are only predicted from intra blocks and it is usually adopted in error-prone scenarios for error-resilience purpose. For comparison, three other transcoding schemes are considered in this experiment: 1) the *Motion-reuse-DCT* transcoder which directly reuses input motion data and performs re-quantization on transform coefficients; 2) the *Motion-reuse-pixel* transcoder which involves fully decoding and re-encoding using input motion data; 3) the *Full-search* transcoder in which the source stream is decoded and then re-encoded with fully R-D optimal motion estimation and mode selection in pixel domain. In all of these transcoders, the MVs and modes in P-frames as well as the modes in intra blocks are reused.

The R-D curves of the transcoding results are given in Fig. 12, while the transcoding speed in terms of processed frames per second for "Foreman" sequence is compared in Fig. 13. At high

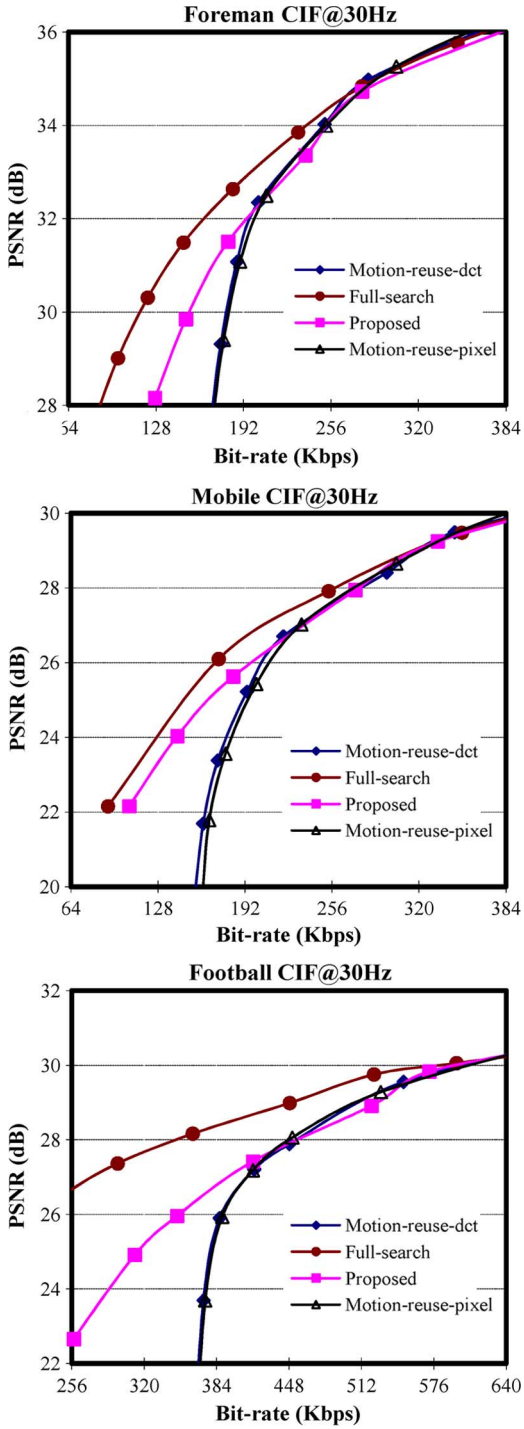

Fig. 12. Performance comparisons among different transcoders for the B-picture transform-domain transcoding.

bit rates, four transcoding schemes have similar results. But at low bit rates, our transcoding scheme achieves an average of 4.0 dB gain over the motion-reuse transcoders in pixel domain and in transform domain as well. Moreover, the complexity of the proposed scheme is only marginally higher than that of the *motion-reuse-DCT* transcoder and is much lower than that of the *full-search* transcoder. Therefore, the proposed transcoding scheme can significantly improve the performance of the hierarchical B-picture transcoding at low bit rates with limited and controllable computation increasing.

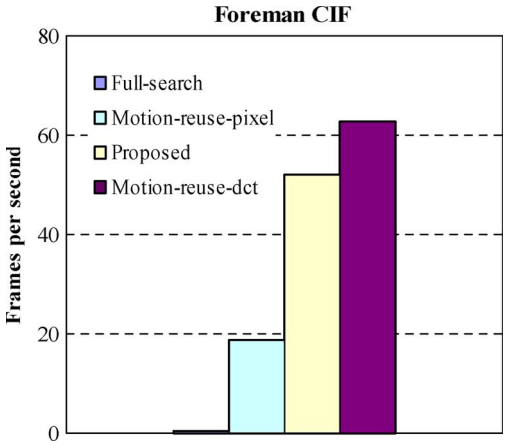

Fig. 13. Complexity comparisons in terms of processed frames per second among different hierarchical B-picture transcoders on Foreman.

VI. Conclusion

In this paper, we propose an RDO model both for the pixel-domain transcoding and the transform-domain transcoding based on H.264. In this model, the distortion, defined as the difference of two predictions which come from two different sets of MVs but the same reference, is demonstrated approximately linear to the MV MSE. So the distortion can be estimated free from SAD/SSD, and the complexity of RDO is significantly reduced as well. Experimental results show that our proposed transcoding schemes in pixel/transform domain along with the proposed RDO method provide better performances in terms of tradeoff between high performance and transcoding speed.

References

[1] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 18–29, Mar. 2003.

[2] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proc. IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.

[3] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: An overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, Oct. 2005.

[4] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264-ISO/IEC 14496 AVC, ITU-T and ISO/IEC JTC1, Mar. 2003.

[5] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[6] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[7] P. Zhang, Y. Lu, Q.-M. Huang, and W. Gao, "Mode mapping method for H.264/AVC spatial downscaling transcoding," in *Proc. Int. Conf. Image Process.*, Oct. 2004, vol. 4, pp. 2781–2784.

[8] P. Zhang, Q.-M. Huang, and W. Gao, "Key techniques of bit-rate reduction for H.264 streams," in *Proc. Pacific-Rim Conf. Multimedia (PCM)*, 2004, pp. 985–992.

[9] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion vector refinement for high-performance transcoding," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 30–40, Mar. 1999.

[10] Y.-P. Tan and H. Sun, "Fast motion re-estimation for arbitrary downsizing video transcoding using H.264/AVC standard," *IEEE Trans. Consum. Electron.*, vol. 50, no. 3, pp. 887–894, Aug. 2004.

[11] C.-H. Li, C.-N. Wang, and T. Chiang, "A fast downsizing video transcoder based on H.264/AVC standard," in *Proc. Pacific-Rim Conf. Multimedia (PCM)*, 2004, pp. 215–223.

[12] H. Shen, X. Sun, F. Wu, and S. Li, "R-D optimal motion estimation for fast H.264/AVC bit-rate reduction," presented at the Picture Coding Symp., Beijing, China, 2006.

[13] H. Shen, X. Sun, F. Wu, H. Li, and S. Li, "A fast downsizing video transcoder for H.264/AVC with R-D optimal mode decision," in *Proc. Int. Conf. Multimedia Expo (ICME)*, Toronto, ON, Canada, Jun. 2006, pp. 2017–2020.

[14] H. Shen, X. Sun, F. Wu, and S. Li, "Rate-distortion optimization for fast hierarchical B-picture transcoding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2006, pp. 5279–5282.

[15] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1029–1041, Aug. 2004.

[16] I. E. G. Richardson, "H.264/MPEG-4 Part 10 White Paper: Transform & Quantization," (2008). [Online]. Available: www.vcodex.com

[17] R. Xiong, J. Xu, F. Wu, and Y.-Q. Zhang, "Layered motion estimation and coding for fully scalable 3D wavelet video coding," in *Proc. Int. Conf. Image Process.*, 2004, vol. 4, pp. 2271–2274.

[18] N. Kamaci, Y. Altunbasak, and R. M. Mersereau, "Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 994–1006, Aug. 2005.

[19] *JSVM 1 Software*, JVT-N22, ITU-T and ISO/IEC JTC1, Jan. 2005.

[20] H.264/AVC Reference Software (2008). [Online]. Available: http://www.iphome.hhi.de/suehring/tml/download/

Huifeng Shen received the B.S. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2005, where he is currently working toward the Ph.D. degree.

His research interests include video compression, video transcoding, and texture compression.

Xiaoyan Sun received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1997, 1999, and 2004, respectively.

She joined Microsoft Research Asia, Beijing, China, as an Associate Researcher in 2003 and has been a Researcher since 2006. She has authored and co-authored over 30 conference and journal papers and submitted several proposals and contributed techniques to MPEG-4 and H.264. Her research interests include video/image compression, video streaming and multimedia processing.

Feng Wu (M'99–SM'06) received the B.S. degree in electrical engineering from Xidian University, China, in 1992, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1996 and 1999, respectively.

He joined in Microsoft Research China, Beijing, China, as an Associate Researcher in 1999. He has been a Researcher since 2001. His research interests include image and video representation, media compression and communication, computer vision, and graphics. He has been an active contributor to ISO/MPEG and ITU-T standards. Some techniques have been adopted by MPEG-4 FGS, H.264/MPEG-4 AVC, and the coming H.264 SVC standard. He served as the Chairman of the China AVS video group from 2002 to 2004 and led the efforts on developing the China AVS video standard 1.0. He has authored or coauthored over 100 conference and journal papers. He has about 30 U.S. patents granted or pending in video and image coding.