Efficient Video Transcoding from H.263 to H.264/AVC Standard with Enhanced Rate Control

Viet-Anh Nguyen and Yap-Peng Tan

School of Electrical & Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

A new video coding standard H.264/AVC has been recently developed and standardized. The standard represents a number of advances in video coding technology in terms of both coding efficiency and flexibility and is expected to replace the existing standards such as H.263 and MPEG-1/2/4 in many possible applications. In this paper we investigate and present efficient syntax transcoding and downsizing transcoding methods from H.263 to H.264/AVC standard. Specifically, we propose an efficient motion vector reestimation scheme using vector median filtering and a fast intraprediction mode selection scheme based on coarse edge information obtained from integer-transform coefficients. Furthermore, an enhanced rate control method based on a quadratic model is proposed for selecting quantization parameters at the sequence and frame levels together with a new frame-layer bit allocation scheme based on the side information in the precoded video. Extensive experiments have been conducted and the results show the efficiency and effectiveness of the proposed methods.

1. INTRODUCTION

The presence of various efficient video coding standards has resulted in a large number of videos produced and stored in different compressed forms [1]. These coding standards compress videos to meet closely the constraints of their target applications, such as available transmission bandwidth, desired spatial or temporal resolution, error resilience, and so forth. Consequently, videos compressed for one application may not be well suited for other applications subject to a set of more restricted constraints, for example, a lower channel capacity or a smaller display screen. To a certain extent, this mismatch in application constraints has hindered efficient sharing of compressed videos among today's heterogeneous networks and devices.

To address such inefficiency, video transcoding has been proposed to convert an existing compressed video to a new compressed video in a different format or syntax [2, 3]. Video transcoding techniques can be broadly classified into homogenous and heterogenous transcodings. Homogeneous transcoding is generally used to reduce the bitrate, frame rate, and/or spatial resolution (downsizing transcoding) so that the processed video can suit better the new application constraints (e.g., small display screen, limited processing resource, or scarce transmission capacity). On the other hand, heterogenous transcoding is used to change the syntax of a compressed video (syntax transcoding) for decoders compliant to a different compression standard, such as the conversion between MPEG-2 and H.263 standards [4]. To meet the requirements of many potential real-time applications, existing video transcoding techniques mostly focus on a few computationally intensive encoding functions (e.g., motion estimation or discrete cosine transform) to speed up the transcoding process. Many also exploit the information extracted from the precoded video [5–7].

Meanwhile, in response to the need of a more efficient video coding technique for diversified networks and applications, H.264/AVC video coding standard has been recently developed and standardized collaboratively by the ITU-T VCEG and the ISO/IEC MPEG standard committees [8]. The standard achieves high coding efficiency by employing a number of new technologies, including multiple reference frames, variable block sizes for motion estimation and compensation, intraprediction coding, 4×4 integer transform, in-loop deblocking filter, and so forth. Empirical studies have shown that H.264/AVC can achieve up to approximately 50% bitrate savings for similar perceived video quality as compared with other existing standards, such as H.263 and MPEG-4. In view of this much improved performance, it is expected that a large number of videos and

devices compliant to the H.264/AVC standard will soon become popular. Hence, there is a need for transcoding precoded videos to H.264/AVC format.

However, due to its new coding features, H.264/AVC is much more different and complex than other existing standards. For example, multiple reference frames and variable block sizes make the motion estimation in H.264/AVC much more complex than that of other standards. Besides motion estimation, intraprediction and coding mode decision in a rate-distortion optimized fashion also increase the coding complexity substantially. Besides, these new features also make accurate rate control more difficult and challenging for both coding and transcoding in H.264/AVC standard [9]. Due to these differences, direct application of existing transcoding techniques may not be efficient and suitable for this new standard.

In this paper, we investigate and propose efficient methods for transcoding H.263 video to H.264/AVC standard by exploiting the new coding features. Specifically, the proposed methods aim to reduce the computational complexity while maintaining acceptable video quality for syntax transcoding and 2 : 1 downsizing transcoding from H.263 to H.264/AVC standard. In a nutshell, the proposed methods include three components, namely fast intraprediction mode selection, motion vector reestimation and intermode selection, and enhanced rate control for H.264/AVC transcoding. The first two components focus on the most computationally intensive parts of the H.264/AVC standard to speed up the transcoding process, while the third component aims to achieve a better video quality by enhancing the rate control with the side information extracted from the precoded video. The experimental results show that the proposed methods can reduce the total encoding time by a factor of 6 and suffer only about 0.35 dB loss in peak-signal-to-noise ratio (PSNR).

The remainder of the paper is organized as follows. Section 2 briefly describes the new H.264/AVC coding features exploited in this paper. Section 3 presents the proposed fast methods for syntax transcoding and downsizing transcoding from H.263 to H.264/AVC standard as well as the enhanced rate control method. The experimental results are shown in Section 4. In Section 5, we conclude the paper by summarizing the main contributions. A preliminary version of this work has been presented in [10].

2. BRIEF OVERVIEW OF H.264/AVC STANDARD

The H.264/AVC standard incorporates a set of new coding features to achieve its high coding efficiency at the cost of substantial increase in complexity. In this section, we summarize the key features, which contribute to the encoder complexity and should be considered in video transcoding to improve the performances in terms of both processing speed and video quality. Interested readers are referred to [11] for a more comprehensive overview of H.264/AVC.

2.1. Coding features

The H.264/AVC standard employs a hybrid coding approach similar to many existing standards but different substantially in terms of the actual coding tools used. Figure 1 shows the block diagram of a typical H.264/AVC encoder.

Like other existing standards, H.264/AVC also employs a block-based motion estimation and compensation scheme to reduce the temporal redundancy in a video bit stream. However, it enhances the performance of motion estimation by supporting multiple reference frames and variable block sizes. Each 16×16 macroblock can be partitioned into 16×16, 16×8, 8×16, and 8×8 samples, and when necessary, each 8×8 block of samples can be further partitioned into 8×4, 4×8, and 4×4 samples, resulting in a combination of seven motion-compensated prediction (MCP) modes (see Figure 2). To attain more precise motion compensation in areas of fine or complex motion, the motion vectors are specified in quarter-pixel accuracy. Furthermore, up to five previously coded frames can be used as references for interframe macroblock prediction. These features make motion estimation in H.264/AVC much more complex compared to that of other existing standards.

In addition, in contrast to previous standards where intraprediction is conducted in the transform domain, the intraprediction in H.264/AVC is formed in the spatial domain based on previously encoded and reconstructed blocks. There are a total of nine possible prediction modes for each 4×4 luma block, four modes for a 16×16 luma block, and four modes for a chroma block, respectively. The number of the intraprediction modes are intrinsically complex and require much computation time [11].

Besides motion estimation and intraprediction, coding mode decision is another main process that increases the computational complexity of a typical H.264/AVC encoder. To attain a high coding efficiency, the H.264/AVC standard software exhaustively examines all coding modes (intra, inter, or skipped) for each macroblock in a rate-distortion (RD) optimized fashion, minimizing a Lagrangian cost function in the form of

$$J = D + \lambda R, \tag{1}$$

where D denotes some distortion measure between the original and the coded macroblock partitions predicted from the reference frames, R represents the number of bits required to code the macroblock difference, and λ is the Lagrange multiplier imposing a suitable rate constraint. To obtain the best coding mode, the encoder in fact performs a real coding process, including prediction and compensation, transformation, quantization, and entropy coding for all inter and intramodes, resulting in a heavy computational load.

2.2. H.264/AVC rate control

The advanced features in H.264/AVC make it difficult and inefficient to employ the existing rate control schemes of other standards. The rate control adopted by the H.264/AVC standard uses an adaptive frame-layer rate control scheme based on a linear prediction model [12].

In the frame-layer rate control, the target buffer bits T_{buf} allocated for the jth frame are determined according to the target buffer level $\text{TBL}(n_j)$, the actual buffer occupancy
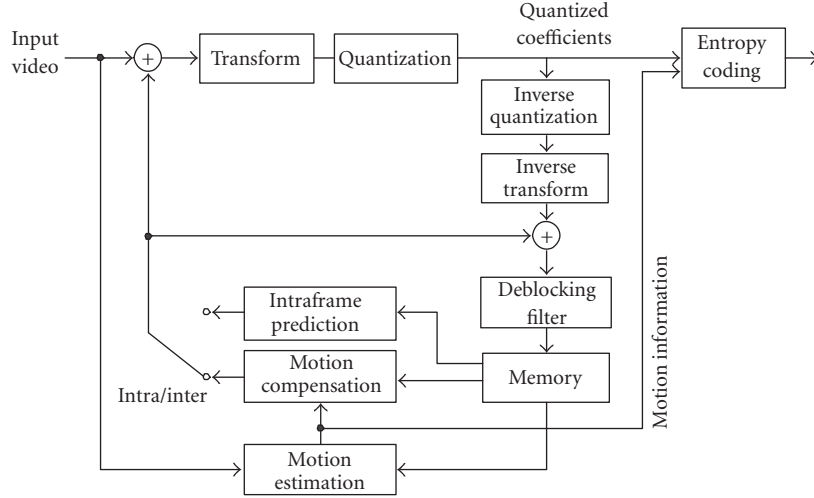
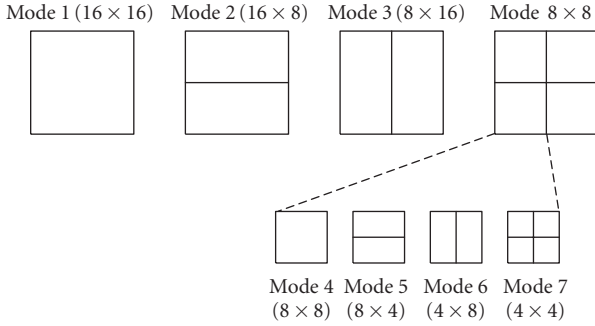FIGURE 1: Block diagram of a typical H.264/AVC encoder.

FIGURE 2: Possible modes for motion-compensated prediction in H.264/AVC.

$B_c(n_j)$, the available channel bandwidth $u(n_j)$, and the frame rate F_r as follows:

$$T_{\text{buf}} = \frac{u(n_j)}{F_r} + \gamma\left(\text{TBL}(n_j) - B_c(n_j)\right), \qquad (2)$$

where γ is a constant and its typical value is 0.75. In addition, the remaining bits are equally allocated to all not-yet-coded frames and the number of bits allocated for each frame is given by

$$T_r = \frac{R_r}{N_r}, \qquad (3)$$

where R_r is the number of remaining bits and N_r is the total number of not-yet-coded frames. Then, the target bit is a weighted combination of T_r and T_{buf},

$$T = \beta \times T_r + (1 - \beta) \times T_{\text{buf}}, \qquad (4)$$

where β is a weighting factor.

A quadratic RD model is used to calculate the corresponding quantization parameter (QP), which is then used for the RD optimization for each macroblock in the current frame. Note that the RD model requires the mean-of-absolute difference (MAD) of the residue error to estimate the QP, which is only available after RD optimized process, thus resulting in a chicken-and-egg problem.

To solve this dilemma, the MAD in the RD model is predicted by a linear model using the actual MAD of the previous frames (refer to [12] for details). However, the linear model assumes the frame complexity varies gradually. If a scene change occurs, the prediction based on the information collected from the previous frames may not be accurate, and in turn it may fail to obtain a suitable QP. Consequently, the number of coding bits for the current frame may not meet the target allocation bits, resulting in quality degradation.

In addition, it should be noted that the first I and P frames in the current group of pictures (GOP) are coded by using the QP given at the GOP layer, in which the starting QP of the first GOP is predefined and the starting QPs of other GOPs are computed based on the QPs of the previous GOP. Thus, an inappropriately predefined starting QP can affect the actual achievable bitrate and video quality. Too small a starting QP would allocate more bits to the first few frames; hence there would not be enough bits for coding other frames to closely meet the target bitrate and inconsistent video quality would result. On the other hand, too large a starting QP would result in a low quality for the first reference frame, which in turn affects the quality of the subsequent frames.

In summary, the advanced coding features in H.264/AVC can provide a better coding efficiency at the cost of increasing complexity. As many potential applications of video transcoding require the video to be transcoded in real time or as fast as possible (e.g., video streaming over heterogenous networks), it is therefore necessary to minimize the complexity of video transcoding without sacrificing much its coding efficiency. In this paper, we focus on the most

TABLE 1: PSNR results (in dB) obtained by the cascaded H.264/AVC recoding approach using four schemes with different combinations of MCP modes and reference frames.

Sequence	Scheme (I)	Scheme (II)	Scheme (III)	Scheme (IV)
Foreman	32.29	33.02	33.17	33.26
Stefan	27.32	27.66	27.95	28.12
News	34.14	34.55	34.82	34.97
Tennis	31.44	31.85	31.96	32.01
Flower	33.47	33.52	33.61	33.65
BBC	29.78	30.73	30.86	30.95
M & D	34.88	35.59	35.67	35.72
Mobile	29.36	29.59	29.66	29.76
Average	31.58	32.06	32.21	32.31

computationally intensive parts of H.264/AVC coding, including intramode prediction, motion estimation, and coding mode decision, to speed up the transcoding process. Furthermore, by using the information available in the precoded video, we further enhance the H.264/AVC rate control to achieve a better quality for the transcoded video.

2.3. Efficient options and modes for transcoding

Before discussing in detail the proposed transcoding methods, it should be noted that a large number and combination of MCP modes and prediction reference frames for each macroblock are possible. Searching over all possible combinations of modes and reference frame options to maximize the overall RD performance is computationally intensive. Moreover, performance analysis conducted by Joch et al. [13] on fourteen common test sequences has shown that more than 80% bit savings gained by exploiting all possible macroblock partitions can be obtained using partitions not smaller than 8×8. Furthermore, when multiple frame prediction is employed, the average bit savings for twelve test sequences are less than 5% and around 20% for the remaining two.

To examine whether the coding performance remains the same for video transcoding using H.264/AVC, we transcoded eight precoded H.263 sequences at 30 frames/s without using B frames (as shown in Table 1) to H.264/AVC at reduced bitrates using the cascaded recoding approach (i.e., the precoded videos were fully decoded and then reencoded using the H.264/AVC standard software). Four schemes using different combinations of MCP modes and reference frames were considered: (I) one mode (mode 1) and one reference frame, (II) four modes (modes 1–4) and one reference frame, (III) all seven modes and one reference frame, and (IV) all seven modes and five reference frames.

The results show that compared with scheme (I), scheme (II) can obtain an average 0.5 dB PSNR improvement. However, the performance gain by using scheme (IV) compared with that of using scheme (II) is only 0.25 dB on average. In addition, by exploiting all partitions smaller than 8×8 with

one reference frame, scheme (III) can obtain only an average 0.15 dB PSNR gain compared with scheme (II). In our view, the much higher computation and memory cost required by exploiting all the possible coding modes and reference frame options cannot justify the small incremental performance gain for video transcoding. Hence, we will limit our proposed H.264/AVC transcoding methods to mainly using four MCP modes (modes 1–4) and one reference frame to minimize the transcoding time.

3. PROPOSED H.263 TO H.264/AVC TRANSCODING METHODS

Figure 3 shows the architecture of the proposed video transcoder. It consists of a typical H.263 decoder followed by a H.264/AVC video encoder. The precoded H.263 video is first decoded by the H.263 decoder and then reencoded by the H.264/AVC video encoder. For downsizing transcoding, the decoded video will be down-sampled before it is transcoded to a H.264/AVC video. In what follows, we present the three key components of our proposed H.264/AVC video transcoding methods: (1) fast intraprediction mode selection, (2) motion vector reestimation and intermode selection, and (3) enhanced rate control.

3.1. Fast intraprediction mode selection

4×4 luma prediction

In intraprediction, the H.264/AVC encoder selects the mode that minimizes the sum-of-absolution difference (SAD) of 4×4 integer-transform coefficients of the difference between the prediction and the block to be coded. Although full search can obtain the optimal prediction mode, it is computationally expensive. Pan et al. [14] propose a fast intraprediction mode selection scheme based on edge direction histogram; however the computation of edge direction introduces additional complexity. Inspired by a key observation that the best prediction mode of a block is most likely in the direction of the dominant edge within that block, we propose a fast intraprediction mode selection scheme based on the coarse edge information obtained from the integer-transform coefficients.

Note that in the DC prediction mode, the residue is computed by offsetting all pixel values of the block to be coded by the same value. Thus, the AC coefficients of the 4×4 integer transform of the residue in the DC prediction mode are the same as the transform coefficients of the block to be coded. Similar to discrete cosine transform (DCT) [15], these integer-transform coefficients can be used to extract some low-level feature information.

Figure 4 shows pictorially the representations for some AC coefficients of the 4×4 integer transform. It can be seen that the value of AC coefficient F_{01} essentially depends upon intensity difference in the horizontal direction between the left-half and the right-half of the block, gauging the strength of vertical edges. Hence, some coarse edge information, such as vertical and horizontal dominant edges, or edge
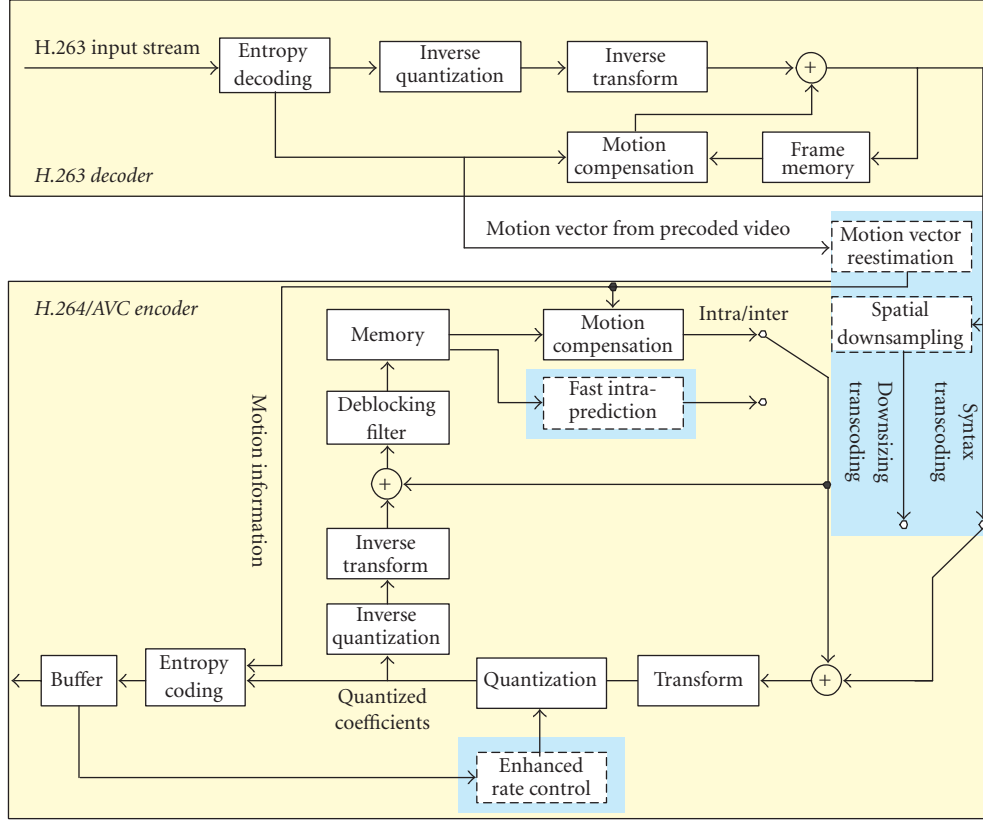
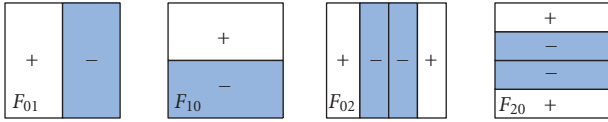FIGURE 3: Block diagram of the proposed video transcoder.

FIGURE 4: Pictorial representation of some 4×4 integer-transform coefficients of the difference between the prediction and the block to be coded in the DC prediction mode.

FIGURE 5: Directions of nine possible intraprediction modes for a 4×4 block.

orientation, can be extracted using these AC measurements in a way similar to that shown in [15] for DCT coefficients. Extending the results obtained in [15], we propose in this paper to estimate the dominant edge orientation by

$$\theta = \tan^{-1} \left(\frac{\sum_{j=1}^{3} F_{0j}}{\sum_{i=1}^{3} F_{i0}} \right), \qquad (5)$$

where θ is the angle of the dominant edge with respect to the horizontal axis and F_{ij}'s are the integer-transform coefficients of a 4×4 block.

Given the angle θ of the dominant edge, we propose to select additional two out of nine intraprediction modes, which have closest orientations to the edge angle θ, for a 4×4 luma prediction. Note that the edge directions of the nine possible prediction modes are shown in Figure 5. Hence, if the angle θ of the dominant edge is between $-26.6°$ and $0°$,

modes 1 and 6 will be selected. Therefore, together with the DC mode, we only need to perform the prediction for three modes instead of nine for a 4×4 block. As the DC mode is always included in 4×4 luma prediction, we can compute (5) using the AC coefficients of 4×4 integer transform of the residue in the DC prediction mode, which are available during the computation of its cost function in intraprediction [11], without incurring much additional computation.

TABLE 2: Average and cumulative percentages of the optimal MV distribution measured at different absolute distances from the new search center in eight test sequences.

Vertical/horizontal distance	Total percentage at different absolute vertical/horizontal distances from the new search center							
	0	1	2	3	4	5	6	7
0	64.8920	7.7059	0.6248	0.4464	0.2391	0.1733	0.1875	0.2691
1	9.3550	5.1418	0.5633	0.2701	0.1336	0.0715	0.0735	0.0884
2	0.7161	0.9548	0.3081	0.1211	0.0586	0.0376	0.0305	0.0267
3	0.4097	0.4086	0.1631	0.1704	0.0685	0.0327	0.0295	0.0277
4	0.2227	0.1856	0.0923	0.0932	0.0828	0.0404	0.0265	0.0236
5	0.1289	0.0908	0.0735	0.0361	0.0564	0.0508	0.0319	0.0227
6	0.1399	0.0852	0.0403	0.0337	0.0235	0.0421	0.0388	0.0269
7	0.1966	0.0821	0.0394	0.0420	0.0217	0.0201	0.0427	0.0459
Average percentage and cumulative percentage of optimal MV distribution at different absolute distances								
Average percentage	64.8920	22.203	3.1671	1.9893	1.1764	0.7919	0.7829	0.9755
Cumulative percentage	64.8920	87.095	90.262	92.251	93.427	94.219	95.002	95.978

Hence, the computational complexity for 4×4 luma prediction can be reduced by a factor of 3 compared with the full search of the best intraprediction mode.

16×16 luma prediction

Similarly, we can obtain the edge orientations of four 8×8 blocks in a macroblock from the DCT coefficients available in the precoded video. Taking the average of these edge orientations gives us the dominant edge orientation in the macroblock. Hence, in addition to the DC prediction mode which is common in homogeneous scenes, we propose to select another one out of three other possible modes based on the dominant edge orientation for a 16×16 macroblock. In this way, we can reduce the complexity of 16×16 luma prediction by a factor of 2.

Note that the fast intraprediction of the proposed transcoder is still conducted in spatial domain. It only makes use of 4×4 integer-transform coefficients and 8×8 DCT coefficients available during transcoding process for estimating the dominant edge direction to reduce the complexity of intramode prediction.

3.2. Motion vector reestimation and intermode selection

To reduce the complexity of video transcoding, many existing methods propose to estimate the new motion vectors (MVs) required for the transcoded video directly from the MVs existing in the precoded video. In this paper, we use the vector median filter, which has been shown to be able to achieve generally the best performance [6], to resample the MVs in the precoded video. The operation of the vector median filter over a set of K corresponding MVs $V = \{mv_1, mv_2, \ldots, mv_K\}$

is given by

$$mv_{\mathrm{VM}} = \arg \min_{mv_j \in V} \sum_{i=1}^{K} ||mv_j - mv_i||_\gamma,$$
$$mv' = \mathcal{S} \times mv_{\mathrm{VM}}, \tag{6}$$

where mv_{VM} denotes the vector median, $|| \cdot ||_\gamma$ the γ-norm for measuring the distance between two MVs, mv' the new MV required, and $\mathcal{S}$ a 2×2 diagonal matrix downscaling the vector median mv_{VM} to suit the reduced frame size in the 2 : 1 downsizing transcoding. Note that in this paper the Euclidean norm ($\gamma = 2$) is adopted for measuring the distance between two MVs.

During the encoding process, the H.264/AVC encoder needs to examine all modes and find the MV of each partition. However, a small number of available MVs for each macroblock in the H.263 precoded video makes it hard to estimate the required MVs accurately. Note that in H.264/AVC standard, the predicted MV from the neighboring macroblocks is used as the MV of the skipped mode. Thus, to enhance the transcoding performance, this predicted MV is also taken into account for estimating the new MVs.

Before we describe our proposed method, let us examine the distribution of the optimal MVs obtained by performing exhaustive search around the precoded and predicted MVs in transcoding eight well-known test sequences (listed in Table 1) consisting of different spatial details and motion contents. Table 2 shows the average and cumulative percentages of the optimal MV distribution around either the precoded or the predicted MV, that is, the one that achieves the smaller SAD is selected as the new search center. For visualization, Figure 6 also shows the distribution of the optimal MVs around the new search center. The results show that most MVs obtained by exhaustive search are centered around the new search center. Specifically, around 87% of
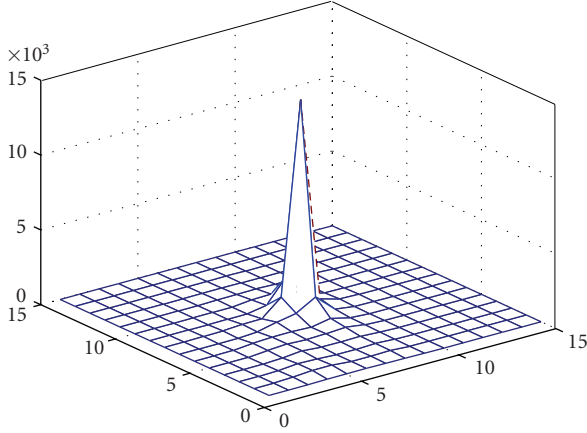
FIGURE 6: Distribution of the MVs obtained by exhaustive search around the precoded MV or the predicted MV from the neighboring macroblocks.

the optimal MVs are enclosed in a 3×3 window area centered around either the precoded or the predicted MV. Based on this empirical study, we propose a scheme for reestimating the new MVs required as follows.

Syntax transcoding

The MV required for each partition of each mode is simply selected from the MV in the precoded video and the predicted MV; the one that achieves the smaller SAD is selected as the new MV.

Downsizing transcoding

The median MV (mv_{VM}) is first obtained from the precoded MVs for each partition of different modes as follows.

Mode 1. The mv_{VM} is the downscaled median MV obtained from the four corresponding MVs in the precoded video (see (6)).

Mode 2. The mv_{VM} of the upper partition is estimated from the downscaled MVs of the two upper corresponding macroblocks; the one that achieves a smaller SAD is selected as the new MV for the upper partition. Similarly, the mv_{VM} for the lower partition is estimated from the downscaled MVs of the two lower corresponding macroblocks.

Mode 3. Similar to mode 2, the mv_{VM}'s of the left and right partition are estimated from the downscaled MVs of the two left and right corresponding macroblocks, respectively.

Mode 8×8. The mv_{VM} for each subpartition in an 8×8 block is simply estimated as the downscaled MV from the corresponding macroblock in the precoded video.

The new MV required for each partition of each mode is then estimated from the mv_{VM} and the MV predicted from neighboring blocks; the one that achieves a smaller SAD will be selected. Note that if a macroblock is intracoded in the

precoded video, the zero MV will be used to reestimate the MVs required.

Since the MVs obtained by exhaustive search are mostly centered within a small window around the reestimated MVs obtained using the above steps, we also propose to refine the reestimated MVs by searching a small diamond pattern centered at the reestimated MVs [16]. To further improve the performance, the refined MVs in integer resolution can be further refined using the default quarter-pixel accuracy in H.264/AVC. To reduce the complexity, we propose to first choose the optimal intermode based on the smallest SAD value obtained by the refined MVs in integer resolution for each mode. Thus, the MVs of only one mode need to perform the quarter-pixel refinement. Furthermore, no RD optimized process is required to choose the best intermode, which can reduce the computational load significantly.

By using MV reestimation, we can reduce the computational complexity for video transcoding. However, during the RD optimized process, the transcoder still needs to make a decision between intra and intermode for each macroblock. It should be noted that the mode decision process of intramode is computationally intensive and may cost five times of that for intermode [17]. Based on our empirical study, we propose to adopt the MV reestimation without using intramode prediction for coding macroblocks in P frames. The reason is that we can reduce the complexity notably without introducing much degradation given that the only information available to the transcoder is the compressed video which is already lossy compressed.

3.3. Enhanced rate control for H.264/AVC transcoding

Rate-quantization ratio model

Both the H.263 and H.264/AVC reference models approximate the relation between the rate and distortion through a quadratic model, in which the number of coding bits is a quadratic function of the quantization step size. Thus, there may be a computable relation between the total number of coding bits in the precoded and transcoded videos.

To confirm, we transcoded the Foreman sequence, which was precoded by H.263 using a constant QP, to H.264/AVC using another fixed QP. Figure 7 shows the relation between the total number of coding bits per frame in the precoded and transcoded videos at different QPs. The figures show that it is likely to have a linear relation between the number of coding bits for each frame in the precoded and transcoded videos. Note that each curve in Figure 7 contains two linear segments, in which the top-right segment representing a greater number of coding bits corresponds to I frames; while the bottom-left segment denoting a smaller number of coding bits corresponds to P frames. It can be seen that the slopes of the two segments are not the same and vary for different QPs, thus suggesting the linear relation could be different for I and P frames and depends on the quantization step sizes of the precoded and transcoded videos.

To justify the above argument, we transcoded five precoded H.263 test sequences to H.264/AVC using different constant QPs. Figure 8 shows the relation between the
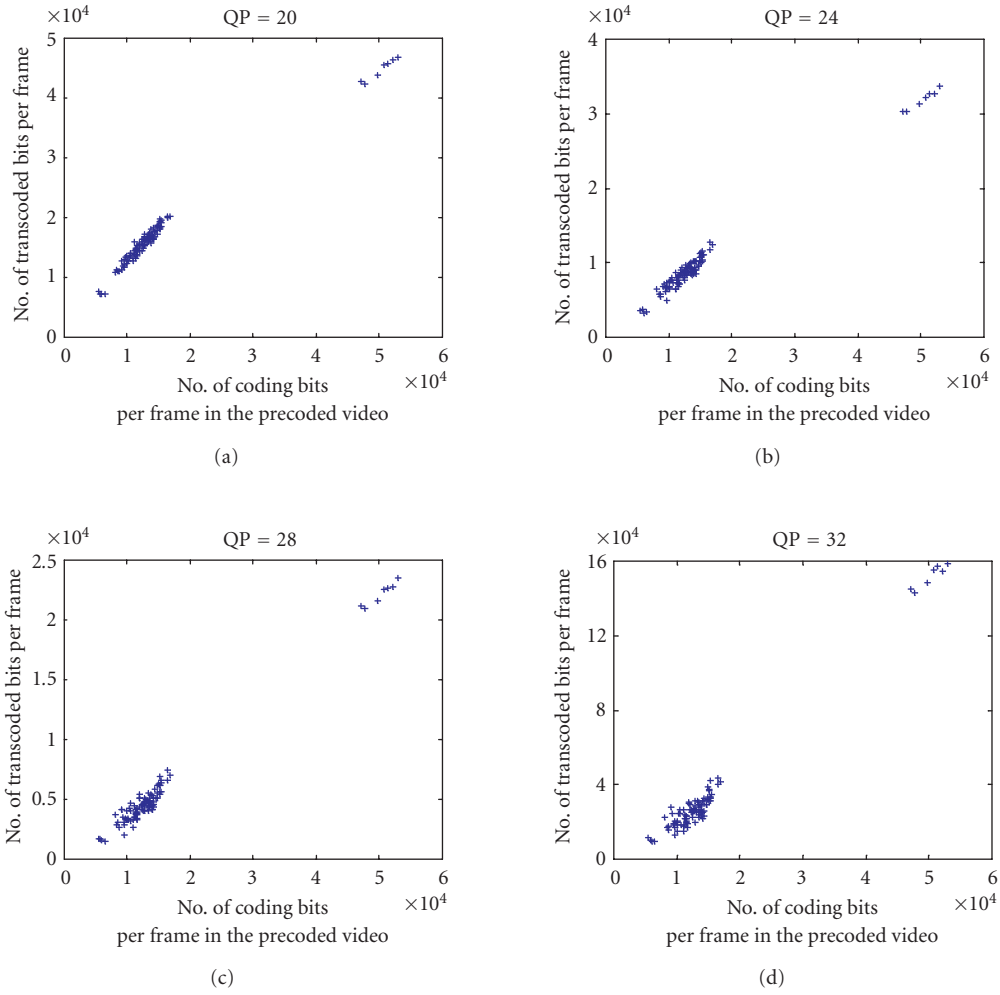
FIGURE 7: Relation between the number of coding bits in precoded and transcoded videos by transcoding using a fixed QP.
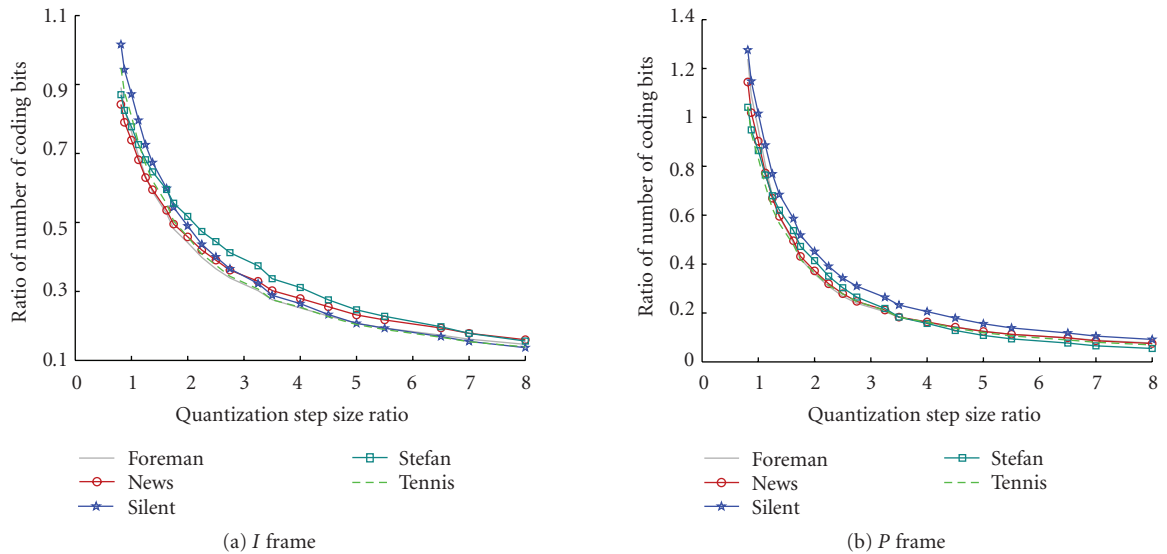

FIGURE 8: Relation between the average ratio of total number of coding bits and the ratio of quantization step sizes in precoded and transcoded videos.

average ratio of total number of coding bits and the quantization step size ratio between the precoded and transcoded videos for I and P frames. The results show that the ratio of total number of coding bits between the precoded and transcoded videos most likely depends on the quantization step size ratio and could be nearly constant for different video contents.

In this paper we propose to use a quadratic model to approximate these relations, which basically follow the trend of the actual curves. Mathematically, the proposed rate-quantization ratio (R_r-Q_r) model is given by

$$\begin{aligned}
\frac{R_t^I}{R_p^I} &= \frac{X_1^I}{(Q_t/Q_p)^2} + \frac{X_2^I}{(Q_t/Q_p)} + X_3^I, \\
\frac{R_t^P}{R_p^P} &= \frac{X_1^P}{(Q_t/Q_p)^2} + \frac{X_2^P}{(Q_t/Q_p)} + X_3^P,
\end{aligned} \tag{7}$$

where $R_p^{I,P}$ and $R_t^{I,P}$ are the total numbers of coding bits, $Q_p^{I,P}$ and $Q_t^{I,P}$ are the quantization step sizes for I and P frames in the precoded and transcoded videos, respectively, and $X_1^{I,P}$, $X_2^{I,P}$, and $X_3^{I,P}$ are the model parameters. The model parameters are empirically obtained by simulation with a large number of video sequences, in which the linear least square method is used to fit the actual curves. Note that the parameters of the model are adaptively updated by using actual data points obtained during the transcoding process to make a better fit for the current video sequence.

Proposed rate control method

(1) Selection of starting QP. In what follows, we propose to determine the good enough starting QP of the sequence or current GOP in order to meet closely the target bitrate. As the quality fluctuation has a negative effect on the subjective video quality, it is desirable to produce a constant quality for the transcoded video. Many experiments have indicated that using constant QP for the entire video sequence typically results in good performance, in terms of both average PSNR and consistent quality [18]. Hence, we will choose the value of the constant QP, which can obtain the transcoded bitrate as close to the target bitrate as possible, as starting QP.

Let Q_t be the quantization step size for transcoding the remaining video in order to have the number of transcoded bits close to the number of remaining bits R_t. By using the proposed model, we can express the total number of transcoded bits with the use of a constant Q_t as

$$R_t = \sum_{k=j}^{N} \left(\frac{X_1}{(Q_t/Q_p^k)^2} + \frac{X_2}{(Q_t/Q_p^k)} + X_3 \right) \times R_p^k, \tag{8}$$

where Q_p^k and R_p^k are the quantization step size and the total number of coding bits of the kth frame in the precoded video, j is the frame number of the first frame in the current GOP, N is the total number of frames, and X_1, X_2, and X_3 are the corresponding model parameters depending on the type (I or P frame) of the kth frame. Hence, Q_t can be obtained by solving the above quadratic equation. The starting QP of the sequence or current GOP is determined as the

nearest integer in the quantization table that corresponds to the quantization step size Q_t.

(2) Allocation of frame bits. As mentioned earlier, H.264/AVC rate control computes the target number of bits per frame by allocating the number of remaining bits to all not-yet-coded frames equally. However, in order to achieve consistently good video quality over the entire sequence, a bit allocation scheme should take into consideration the frame complexity. The basic idea is to allocate fewer bits to less complex frames in order to save more bits for more complex frames. In this paper, we use the number of coding bits and quantization step size in the precoded video to measure the complexity S_k of the kth frame as

$$S_k = R_p^k \times Q_p^k. \tag{9}$$

Hence, instead of allocating bits equally as (3), we propose to allocate the number of remaining bits to all not-yet-coded frames proportionally according to the frame complexity. Thus, the number of bits allocated for the kth frame T_r^k can be computed as

$$T_r^k = R_r \times \frac{S_k}{\sum_{i=k}^{N} S_i}. \tag{10}$$

The final target bitrate is then computed using (4).

(3) Determination of frame QP. After target bit allocation, it is important to determine the corresponding QP to meet exactly the target bit budget. However, the RD model in the existing rate control scheme may fail to determine the correct QP due to inaccurate prediction of MAD in the event of abrupt change in frame complexity. In this paper, we propose to use the R_r-Q_r model to determine the QP at frame level.

Similar to (8), the quantization step size Q_t^k for the kth frame can be easily determined by solving

$$T_t^k = \left(\frac{X_1}{(Q_t^k/Q_p^k)^2} + \frac{X_2}{(Q_t^k/Q_p^k)} + X_3 \right) \times R_p^k, \tag{11}$$

where T_t^k is the target number of bits for the kth frame obtained from (4).

4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed transcoding methods, our test sequences include eight popular CIF resolution sequences, as shown in Table 3, which were precoded by using the test model 8 (TMN8) H.263 encoder [19]. In our simulation, the proposed transcoding methods were implemented on the reference software H.264/AVC JM 7.4 [20]. For each test sequence, we set the frame rate to 30 frames/s and selected the appropriate bitrate so that there was no skipped frame in the precoded and transcoded videos. For performance comparison, we kept the bitrate constant when transcoding each sequence using different methods. The

TABLE 3: PSNR results and encoding times obtained by transcoding H.263 sequences using the cascaded H.264/AVC recoding (RC) method, the MV reestimation method proposed in Section 3.2, with or without quarter-pixel refinement (refn.).

(a) PSNR (dB)

Sequence	Transcoded frame size	Scheme (I)		Scheme (II)		H.264/AVC RC
		No refn.	Refn.	No refn.	Refn.	Refn.
Foreman	352×288	34.94	36.43	34.42	36.12	36.67
	176×144	31.57	32.74	31.35	32.61	33.08
M & D	352×288	39.83	40.60	39.67	40.51	40.66
	176×144	37.17	38.05	37.13	38.01	38.12
News	352×288	38.64	39.33	38.33	39.17	39.50
	176×144	34.27	35.10	34.20	35.04	35.32
Silent	352×288	35.08	35.36	34.90	35.26	35.43
	176×144	33.67	34.09	33.55	34.02	34.22
Stefan	352×288	28.28	31.55	27.48	31.05	31.86
	176×144	26.80	28.83	26.46	28.66	29.36
Tennis	352×288	31.12	31.47	30.82	31.33	31.63
	176×144	35.02	35.38	34.88	35.31	35.66
Mobile	352×288	25.97	27.88	25.66	27.83	28.16
	176×144	34.33	35.41	34.31	35.41	35.60
Flower	352×288	29.05	29.81	29.00	29.80	29.90
	176×144	30.84	31.44	30.84	31.45	31.48

(b) Total encoding time (s)

Sequence	Transcoded frame size	Scheme (I)		Scheme (II)		H.264/AVC RC
		No refn.	Refn.	No refn.	Refn.	Refn.
Foreman	352×288	955	1144	220	341	1924
	176×144	217	270	54	82	462
M & D	352×288	949	1092	210	319	1805
	176×144	228	277	57	77	449
News	352×288	997	1130	215	326	1866
	176×144	231	285	55	79	452
Silent	352×288	1082	1214	227	339	1944
	176×144	243	288	60	85	483
Stefan	352×288	948	1134	224	343	1904
	176×144	240	285	62	86	488
Tennis	352×288	1341	1416	271	432	2577
	176×144	269	298	64	85	496
Mobile	352×288	1416	1588	353	488	2551
	176×144	400	451	97	117	642
Flower	352×288	1158	1291	239	342	2402
	176×144	278	306	60	82	475

GOP of each precoded and transcoded sequence consisted of one *I* frame followed by 14 *P* frames. During downsizing transcoding, each precoded frame was reconstructed and downsized in spatial domain using bicubic interpolation. To suppress aliasing artifacts, a typical Gaussian-type lowpass filter was also applied prior to the downsizing operation. For objective comparison, the PSNR of each transcoded video was computed with respect to the original uncompressed

video with downscaling (for downsizing transcoding) or without downscaling (for syntax transcoding) to the same frame size.

In the first set of experiments, eight test sequences precoded in H.263 were transcoded to H.264/AVC using only the MV reestimation method proposed in Section 3.2 with four MCP modes (modes 1–4) and one reference frame to compare with the cascaded recoding (RC) approach using seven MCP modes and one reference frame. For comparison, two schemes of different mode options for coding macroblocks in P frames were considered: (I) using both intra and intermodes and (II) using only intermode. Table 3 shows the PSNR and complexity results in terms of total encoding time based on our implementation. The results show that incurring much lower computational costs, both schemes can perform comparably to the H.264/AVC RC scheme. Specifically, the average PSNR results obtained by the proposed schemes are only about 0.35 dB inferior to that obtained by the H.264/AVC RC scheme both with and without downscaling, while the total encoding time of scheme (II) is reduced by a factor of about 6 compared with that of the H.264/AVC RC scheme. It should be noted that by using quarter-pixel refinement, we can achieve about 0.9 dB and 1.3 dB improvement in PSNR both with and without downscaling, respectively, in both schemes. In addition, without using intramode in P frames, scheme (II) can reduce the computational cost substantially while the performance is only a little worse than that of scheme (I). Furthermore, compared with scheme (I) (without quarter-pixel refinement), scheme (II) (with quarter-pixel refinement) not only is much less computationally expensive, it can also obtain an average of more than 1 dB performance gain in PSNR.

In another set of experiments, we repeated the first experiment by using both transcoding methods proposed in Sections 3.1 and 3.2. Similar to the first experiment, both schemes with and without using intramode in P frames were considered, and denoted as scheme (I') and scheme (II'). Note that the only difference between these schemes and those in Table 3 is that the proposed intraprediction mode selection scheme (IPMS) was used to find the best mode in intraprediction instead of the full search IPMS. Table 4 shows the average PSNR results and total encoding times. Observably, using the proposed IPMS with fewer intraprediction modes for transcoding (scheme (I') and scheme (I")) can reduce the encoding complexity significantly compared with the full search IPMS (scheme (I) and scheme (II)) while introducing only about 0.1 dB loss in PSNR.

Figure 9 shows the frame-to-frame PSNR results of the News sequence obtained by using scheme (I') and scheme (II') in comparison with the H.264/AVC RC scheme. It should be noted that the proposed method can perform modestly inferior compared to the cascaded H.264/AVC RC scheme and the differences are uniformly distributed over the entire sequence. For visual comparison, Figure 10 shows sample frames of the Foreman sequence obtained by the proposed method. The figures show that the proposed method can achieve a good perceived video quality, in terms of sharpness and blocking artifacts, compared with cascaded

recoding method, and perform visually about the same as the cascaded recoding scheme.

To evaluate the performance of the proposed rate control method when used together with the proposed fast transcoding methods, the Foreman sequence was transcoded to H.264/AVC using the proposed R_r-Q_r model to determine the starting QP at different target bitrates that were generated by transcoding with different constant QPs. The simulation shows that the proposed model is able to choose the constant QP, which was used to generate each target bitrate, as the starting QP. Figure 11 shows the PSNR results and the difference between the target and achieved bitrates for various starting QPs. Obviously, the starting QPs in the bottom-right area are preferred, where the differences between the target and achieved bitrates are close to zero and the PSNRs are high enough. The results show that by using the proposed method, the transcoder can determine a reasonably good starting QP (star marker) in order to meet closely the given target bitrates and achieve good video quality. Furthermore, the proposed method can provide more consistent video quality in terms of low PSNR fluctuation (measured by the standard deviation (σ) of the PSNR results over each entire sequence) as shown in Table 5. Note that Table 5 tabulates the numerical results of using several QP values around the selected one (bottom-right area in Figure 11); these QPs generally result in too large a difference between target and achieved bitrates or too low PSNR.

In the last set of experiments, we transcoded six test sequences at QCIF resolution and 15 frames/s by the cascaded H.264/AVC RC approach using seven MCP modes and one reference frame and the fast transcoding methods proposed in Sections 3.1 and 3.2 with both existing H.264/AVC and proposed rate control methods using four MCP modes (modes 1–4) and one reference frame. The results in Table 6 show that the standard deviation of the PSNR performance obtained using the H.264/AVC RC method is slightly better than that achieved by using the proposed fast transcoding methods with existing H.264/AVC rate control. However, by using the proposed rate control method for transcoding, the quality of transcoded video can be further enhanced. Specifically, compared with the H.264/AVC RC method, the transcoded video obtained by using the proposed rate control method can meet the target bitrate more accurately; furthermore, the standard deviation of the PSNR performance is lower than that obtained by the H.264/AVC RC method, which implies a more consistent video quality over the entire sequence.

Figure 12 shows the frame-to-frame PSNR results of the Foreman sequence obtained by the H.264/AVC RC method and the proposed fast transcoding methods together with the enhanced rate control. Not surprisingly, the fluctuation of PSNR obtained by transcoding with the proposed rate control method is less than that of the H.264/AVC RC method. This can be explained by the fact that our method allocates bits per frame based on the frame complexity. In addition, we used the proposed R_r-Q_r model to determine more accurate QP for each frame rather than using MAD prediction that can be inaccurate as discussed before. To see that, Figure 13

TABLE 4: PSNR results and encoding times obtained by transcoding H.263 sequences using the cascaded H.264/AVC recoding (RC) method and the proposed H.264/AVC transcoding methods in Sections 3.1 and 3.2, with or without quarter-pixel refinement (refn.).

(a) PSNR (dB)

Sequence	Transcoded frame size	Scheme (I')		Scheme (II')		H.264/AVC RC
		No refn.	Refn.	No refn.	Refn.	Refn.
Foreman	352×288	34.82	36.35	34.39	36.08	36.67
	176×144	31.46	32.65	31.25	32.52	33.08
M & D	352×288	39.78	40.57	39.63	40.49	40.66
	176×144	37.10	37.97	37.06	37.92	38.12
News	352×288	38.52	39.24	38.28	39.09	39.50
	176×144	34.10	34.90	34.06	34.88	35.32
Silent	352×288	35.02	35.32	34.87	35.24	35.43
	176×144	33.55	33.96	33.46	33.91	34.22
Stefan	352×288	28.10	31.47	27.42	30.98	31.86
	176×144	26.68	28.76	26.36	28.60	29.36
Tennis	352×288	31.06	31.43	30.79	31.30	31.63
	176×144	34.97	35.35	34.83	35.26	35.66
Mobile	352×288	25.87	27.83	25.63	27.80	28.16
	176×144	34.29	35.38	34.28	35.38	35.60
Flower	352×288	28.99	29.75	28.96	29.75	29.90
	176×144	30.72	31.30	30.71	31.32	31.48

(b) Total encoding time (s)

Sequence	Transcoded frame size	Scheme (I')		Scheme (II')		H.264/AVC RC
		No refn.	Refn.	No refn.	Refn.	Refn.
Foreman	352×288	597	752	193	314	1924
	176×144	138	162	44	72	462
M & D	352×288	584	707	185	292	1805
	176×144	135	166	43	73	449
News	352×288	611	749	185	295	1866
	176×144	135	166	42	67	452
Silent	352×288	656	773	192	306	1944
	176×144	146	178	43	75	483
Stefan	352×288	586	745	200	303	1904
	176×144	146	178	49	77	488
Tennis	352×288	641	767	194	303	2577
	176×144	160	188	49	77	496
Mobile	352×288	726	874	236	359	2551
	176×144	278	259	74	102	642
Flower	352×288	692	788	199	302	2402
	176×144	157	198	48	72	475

plots the number of bits allocated per frame and the actual coding bits obtained using the QPs, which were determined by existing H.264/AVC model and proposed R_r-Q_r model. As can be seen, the proposed model can obtain the number of actual coding bits very close to the number of allocation bits.

5. CONCLUSION

We have proposed in this paper an efficient method for H.263 to H.264/AVC video transcoding. Besides using a vector median filter for motion reestimation, we have also proposed
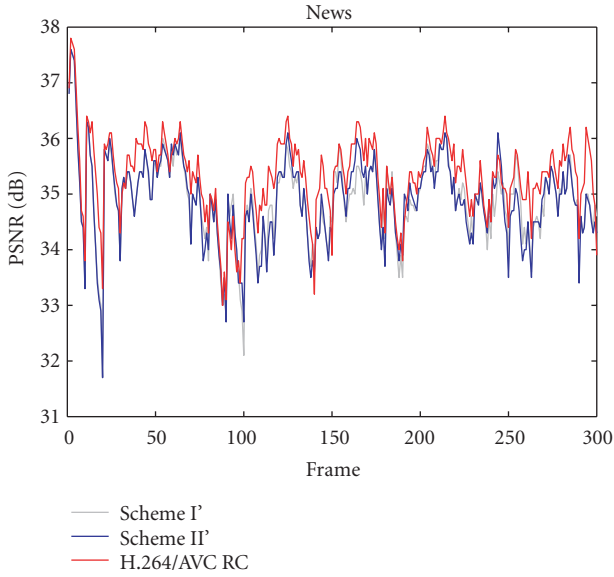
Figure 9: Frame-to-frame PSNR results of the News sequence obtained by the proposed H.264/AVC transcoding methods and the H.264/AVC recoding (RC) method.
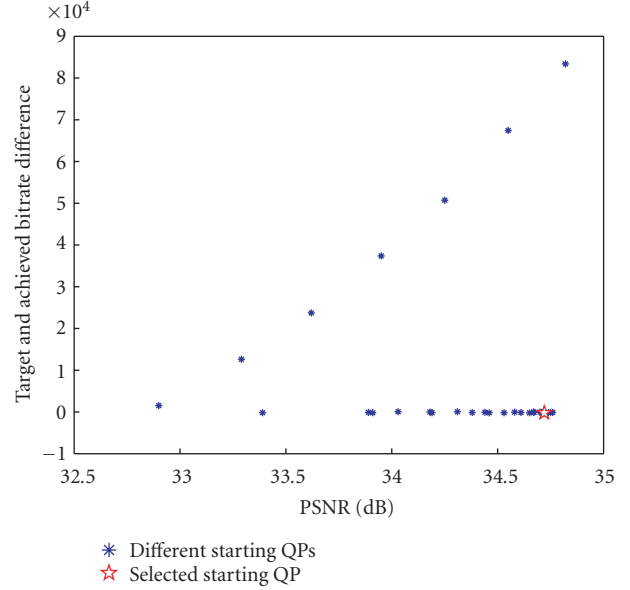

Figure 11: PSNR results (in dB) and the differences between the target and achieved bitrates obtained by transcoding the Foreman sequence using various starting QPs at a given target bitrate.

Original precoded frame (37.59 dB)

(a)

Scheme I'(36.48 dB)

(b)

Scheme II'(36.27 dB)

(c)

H.264/AVC RC (36.72 dB)

(d)

Figure 10: Sample frames from the Foreman sequence transcoded from the precoded H.263 video by the proposed syntax transcoding methods with quarter-pixel refinement and the cascaded H.264/AVC recoding (RC) method.

Table 5: PSNR, standard deviations of the PSNR results (σ), and achieved bitrate for the Foreman sequence using different starting QPs at a given target bitrate. Note that the starting QP obtained by the proposed model is 28.

Starting QP	Target bitrate (kbps)	Achieved bitrate (kbps)	PSNR (dB)	σ
16	84.83	86.36	32.90	3.51
20	84.83	85.55	34.44	1.56
24	84.83	84.67	34.74	0.95
28	84.83	84.70	34.72	0.88
32	84.83	84.62	34.65	0.86
36	84.83	84.18	34.38	1.21

a fast intraprediction mode selection scheme based on the coarse edge information obtained from integer-transform coefficients. In addition, an enhanced rate control method is proposed to improve the transcoded video quality. The proposed rate control method uses a quadratic model for selecting quantization parameters at the sequence and frame levels together with a new frame-layer bit allocation scheme based on the side information from the precoded video. The experimental results show the accuracy of the model and the effectiveness of the proposed methods. In particular, the PSNR obtained by the proposed methods is only about 0.35 dB inferior to that obtained by the cascaded H.264/AVC recoding scheme, while the total transcoding time can be reduced by a factor of 6. Furthermore, the proposed rate control method can meet the target bitrate more accurately and provide more consistent video quality compared with that of existing H.264/AVC rate control scheme.

TABLE 6: PSNR results (in dB), standard deviations of the PSNR results (σ), and actual bitrates obtained by H.264/AVC recoding (RC) method, the proposed fast transcoding methods in conjunction with the existing, and the proposed H.264/AVC rate control methods.

| Sequence | Target bitrate (kbps) | Proposed fast transcoding methods | | | | | | H.264/AVC cascaded recoding | | |
		Existing rate control			Proposed rate control					
		Actual bitrate (kbps)	PSNR (dB)	σ	Actual bitrate (kbps)	PSNR (dB)	σ	Actual bitrate (kbps)	PSNR (dB)	σ
Foreman	65.17	64.87	33.81	1.015	65.15	33.83	0.652	64.58	34.22	0.923
	111.05	111.80	35.52	0.792	111.17	35.57	0.543	111.72	35.95	0.712
News	31.89	32.81	32.92	1.167	31.92	32.95	0.915	32.75	33.27	1.095
	55.46	56.38	35.67	1.093	55.53	35.60	0.835	56.41	36.06	0.964
Silent	60.38	60.74	34.58	0.787	60.36	34.64	0.493	60.59	34.79	0.792
	109.59	110.60	36.51	0.597	109.64	36.56	0.326	108.62	36.78	0.543
Stefan	140.28	140.73	29.03	1.433	140.34	29.11	0.875	140.84	29.75	1.355
	292.04	292.98	32.34	1.281	292.19	32.30	0.613	292.76	33.13	1.153
Tennis	73.49	74.11	30.55	2.191	73.62	30.57	1.593	72.89	30.84	2.018
	136.60	136.92	32.81	1.712	136.58	32.87	1.033	136.89	33.14	1.582
Mobile	194.44	195.54	28.47	1.132	194.66	28.54	0.847	195.32	28.66	1.087
	388.00	388.54	31.03	0.984	388.13	31.17	0.652	388.61	31.25	0.941
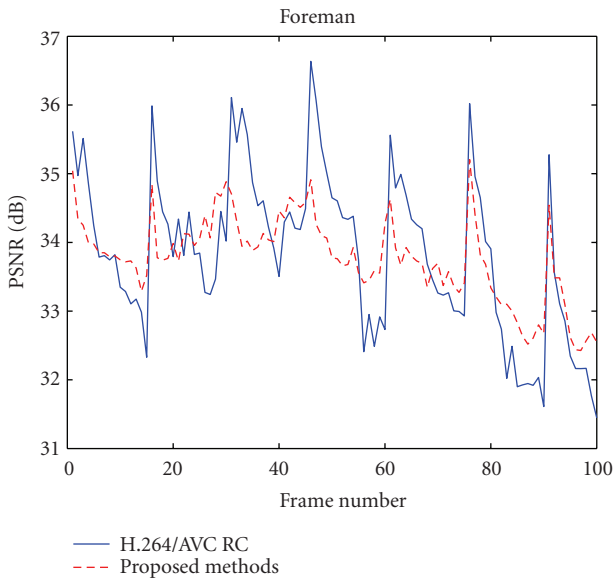
FIGURE 12: PSNR performance (in dB) of the Foreman sequence obtained by the H.264/AVC recoding (RC) method and the proposed fast transcoding methods together with the enhanced rate control.
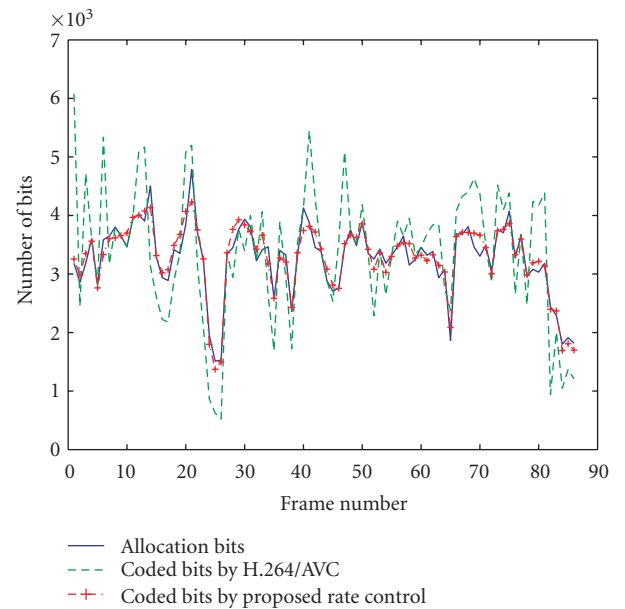

FIGURE 13: Number of bits allocated per frame and actual coding bits obtained using the QP determined by H.264/AVC method and the proposed R_r-Q_r model.

REFERENCES

[1] B. Haskell, P. G. Howard, Y. LeCun, et al., "Image and video coding - emerging standards and beyond," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 814–837, 1998.

[2] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, 2003.

[3] N. Bjork and C. Christopoulos, "Transcoder architecture for video coding," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 1, pp. 88–98, 1998.

[4] T. Shanableh and M. Ghanbari, "Transcoding of video into different encoding formats," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, pp. 1927–1930, Istanbul, Turkey, June 2000.

[5] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video downscaling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 6, pp. 929–936, June 1999.

[6] Y.-P. Tan, H. Sun, and Y. Liang, "On the methods and applications of arbitrary downsizing video transcoding," in *IEEE International Conference on Multimedia and Expo*, pp. 609–612, Lausanne, Switzerland, August 2002.

[7] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Transaction on Multimedia*, vol. 2, no. 2, pp. 101–110, 2000.

[8] T. Wiegand and G. Sullivan, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," March 2003.

[9] J. Ostermann, J. Bormans, P. List, et al., "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[10] V.-A. Nguyen and Y.-P. Tan, "Efficient H.263 to H.264/AVC video transcoding using enhanced rate control," in *IEEE International Conference on Image Processing (ICIP '05)*, Genoa, Italy, September 2005.

[11] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, 2003.

[12] Z. G. Li, F. Pan, K. P. Lim, G. N. Feng, X. Lin, and S. Rahardja, "Adaptive basic unit layer rate control for JVT, JVT-G012-r1," in *7th Meeting, Pattaya II*, Bangkok, Thailand, March 2003.

[13] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of the ITU-T draft H.26L video coding standard," in *12th International Packet Video Workshop*, Pittsburgh, Pa, USA, April 2002.

[14] F. Pan, X. Lin, and R. Susanto, "Fast intra mode decision algorithm for H.264/AVC video coding," in *IEEE International Conference on Image Processing (ICIP '04)*, Singapore, October 2004.

[15] B. Shen and I. K. Shethi, "Direct feature extraction from compressed images," in *Storage and Retrieval for Image and Video Databases IV*, vol. 2670 of *Proceedings of SPIE*, pp. 404–414, San Jose, Calif, USA, February 1996.

[16] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast blockmatching motion estimation," *IEEE Transaction on Image Processing*, vol. 9, pp. 287–290, 2000.

[17] B. Jeon and J. Lee, "Fast mode decision for H.264," ITU-T VCEG (Q.6/16), July 2003.

[18] B. Xie and W. Zeng, "Sequence-based rate control for constant quality video," in *IEEE International Conference on Image Processing (ICIP '02)*, vol. 1, pp. 77–80, Rochester, NY, USA, September 2002.

[19] ITU-T/SG15, "Video Codec Test Model TMN-8," Portland, June 1997.

[20] JM Reference Software Version 7.4, http://iphome.hhi.de/suehring/tml/download.

Viet-Anh Nguyen was born in Vietnam in 1980. He received the B.S. degree in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2004, where he is currently working toward the Ph.D. degree. His current research interests include image and video processing, pattern recognition, computer vision, and color imaging. He received the First Prize in the National Informatics Olympiad in Vietnam in 1998. In 2004, he was awarded the Singapore Millennium Foundation scholarship for his postgraduate study.

Yap-Peng Tan received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1993, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1995 and 1997, respectively, all in electrical engineering. He was the recipient of an IBM Graduate Fellowship from IBM T. J. Watson Research Center, Yorktown Heights, NY, from 1995 to 1997 and was with Intel and Sharp Labs of America from 1997 to 1999. In November 1999, he joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is currently an Associate Professor and Head of the Division of Information Engineering. His current research interests include image and video processing, content-based multimedia analysis, computer vision, and pattern recognition. He is the principal inventor/coinventor of 13 USpatents in the areas of image and video processing. He is an Editorial Board Member of EURASIP Journal on Applied Signal Processing and a Member of the IEEE Circuits and Systems Society's Technical Committee on Visual Signal Processing and Communications.