

New Architecture for Dynamic Frame-Skipping Transcoder

Kai-Tat Fung, Yui-Lam Chan, and Wan-Chi Siu, *Senior Member, IEEE*

Abstract—Transcoding is a key technique for reducing the bitrate of a previously compressed video signal. A high transcoding ratio may result in an unacceptable picture quality when the full frame rate of the incoming video bitstream is used. Frame skipping is often used as an efficient scheme to allocate more bits to the representative frames, so that an acceptable quality for each frame can be maintained. However, the skipped frame must be decompressed completely, which might act as a reference frame to nonskipped frames for reconstruction. The newly quantized discrete cosine transform (DCT) coefficients of the prediction errors need to be re-computed for the nonskipped frame with reference to the previous nonskipped frame; this can create undesirable complexity as well as introduce re-encoding errors. In this paper, we propose new algorithms and a novel architecture for frame-rate reduction to improve picture quality and to reduce complexity. The proposed architecture is mainly performed on the DCT domain to achieve a transcoder with low complexity. With the direct addition of DCT coefficients and an error compensation feedback loop, re-encoding errors are reduced significantly. Furthermore, we propose a frame-rate control scheme which can dynamically adjust the number of skipped frames according to the incoming motion vectors and re-encoding errors due to transcoding such that the decoded sequence can have a smooth motion as well as better transcoded pictures. Experimental results show that, as compared to the conventional transcoder, the new architecture for frame-skipping transcoder is more robust, produces fewer requantization errors, and has reduced computational complexity.

Index Terms—Compressed domain processing, DCT-based transcoder, frame skipping, rate control, video transcoding.

I. INTRODUCTION

WITH the advance of video compression and networking technologies, networked multimedia services, such as multipoint video conferencing, video on demand and digital TV, are emerging [1]–[7]. A video server may have to provide quality support services to heterogeneous clients or transmission channels. It is in this scenario that the video server should have the capability of performing transcoding [8]–[14], which is regarded as a process of converting a previously compressed video bitstream into a lower bitrate bitstream without modifying its original structure.

Manuscript received March 14, 2001; revised March 10, 2002. This work was supported by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University. K. T. Fung was supported by research studentships provided by Hong Kong Polytechnic University. Dr. Y. L. Chan was supported by Hong Kong Polytechnic University under its research fellowship scheme. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Luis Torres.

The authors are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: enktfung@eie.polyu.edu.hk; enylchan@polyu.edu.hk; enwcsiu@polyu.edu.hk).

Publisher Item Identifier 10.1109/TIP.2002.800890.

One straightforward approach for implementing transcoding is to cascade a decoder and an encoder [8]–[11], commonly known as pixel-domain transcoding. The incoming video bitstream is decoded in the pixel domain, and the decoded video frame is re-encoded at the desired output bitrate according to the capability of the clients' devices and the available bandwidth of the network. This involves a high processing complexity, memory, and delay. As a consequence, some information reusing approaches [9], [10] have been proposed. For example, motion vectors extracted from the incoming bitstream after decoding can be used to reduce the complexity of the transcoding significantly. In addition, the video quality of the pixel-domain transcoding approach suffers from its intrinsic double-encoding process, which introduces additional degradation.

In recent years, discrete cosine transform (DCT) domain transcoding was introduced [12]–[14], under which the incoming video bitstream is partially decoded to form the DCT coefficients and downsampled by the requantization of the DCT coefficients. Since DCT-domain transcoding is carried out in the coded domain where complete decoding and re-encoding are not required, the processing complexity is significantly reduced. The problem with this approach, however, is that the quantization errors will accumulate, and a prediction memory mismatch at the decoder will cause poor video quality. This phenomenon is called "drift" degradation, which often results in an unacceptable video quality. Thus, several techniques for eliminating "drift" degradation [12]–[14] have been proposed. DCT-domain transcoding is a very attractive approach for many video applications. However, it is impossible to achieve the desired output bitrate by performing only requantization. In other words, if the bandwidth of the outgoing channel is not enough to allocate bits with requantization, frame skipping is a good strategy for controlling the bitrate and maintaining the picture quality within an acceptable level. It is difficult to perform frame skipping in the DCT-domain since the prediction errors of each frame are computed from its immediate past frames. This means that the incoming quantized DCT coefficients of the residual signal are no longer valid because they refer to the frames which have been dropped. This problem has not been fully considered in the literature. However, several frame-skipping techniques in pixel domain for bitrate reduction of compressed video have been devised in recent years [9]–[11]. For instance, frame-skipping transcoder proposed in [9], [10] made use of the motion vector refinement scheme when the frame-rate conversion is needed. The refinement scheme suggested a forward dominant vector selection (FDVS) method to compose an outgoing motion vector from the incoming

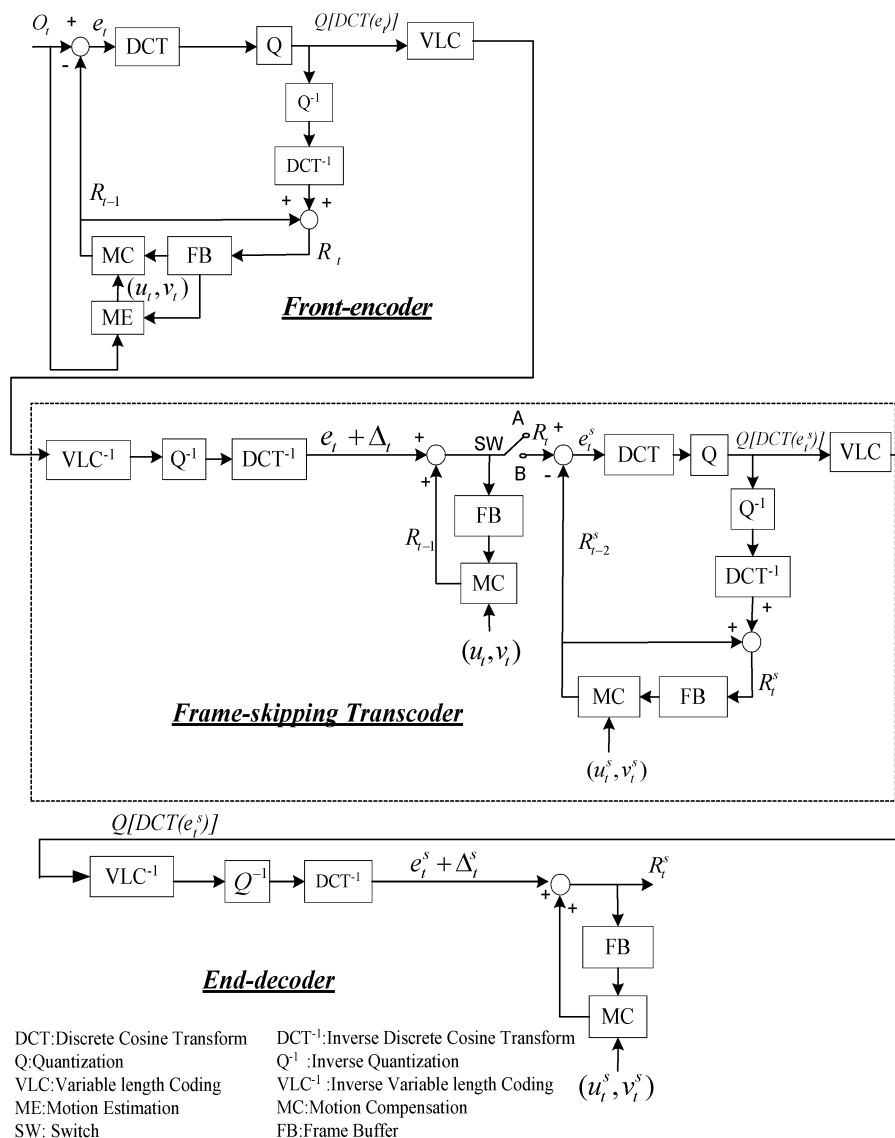


Fig. 1. Frame-skipping transcoder in pixel-domain.

motion vectors of the skipped frames. In [11], a frame control scheme was proposed to dynamically adjust the number of skipped frames according to the accumulated magnitude of motion vectors such that the transcoded sequence can present a much smoother motion. These techniques are useful for frame-skipping transcoders in the pixel-domain. In this paper, we provide a computationally efficient solution to perform frame skipping in a transcoder, mainly in the DCT-domain, to avoid the complexity and the quality degradation arising from pixel-domain transcoding. In addition, a frame-skipping control scheme with dynamic behavior is proposed, which can adaptively skip the unnecessary frames according to the motion information and the re-encoding errors due to transcoding. As a result, our proposed frame-skipping transcoder which has an architecture of low-complexity can provide a smoother and better transcoded sequence.

The organization of this paper is as follows. Section II of this paper presents an in-depth study of re-encoding errors in the frame-skipping transcoder. The architecture of the

TABLE I
SWITCH POSITION FOR DIFFERENT MODES OF THE
PIXEL-DOMAIN TRANSCODER

Frame skipping mode	SW Position
Skipped frame	A
Non-skipped frame	B

proposed dynamic frame-skipping transcoder is then described in Section III. Simulation results are presented in Section IV. Finally, some concluding remarks are provided in Section V.

II. FRAME-SKIPPING IN PIXEL-DOMAIN TRANSCODING

Fig. 1 shows the structure of a conventional frame-skipping transcoder in pixel-domain [8]–[10]. In the front encoder, the motion vector, mv_t , for a macroblock with $N \times N$ pixels in frame O_t , the current frame, is computed [15]–[20] by searching for the best matched macroblock within a search window S

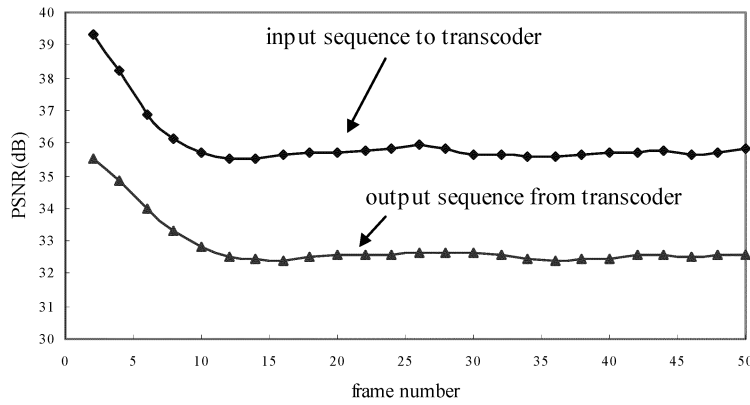


Fig. 2. Quality degradation of conventional frame-skipping transcoder for the “Salesman” sequence. The PSNR of the frame-skipping pictures is plotted to compare with that of the same pictures which used directly a decoder without a transcoder.

in the previous reconstructed frame, R_{t-1} , and it is obtained as follows:

$$mv_t = (u_t, v_t) = \arg \min_{(m, n) \in S} SAD(m, n) \quad (1)$$

$$SAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |O_t(i, j) - R_{t-1}(i + m, j + n)| \quad (2)$$

where m and n are the horizontal and vertical components of the displacement of a matching macroblock, $O_t(i, j)$ and $R_{t-1}(i, j)$ represent a pixel in O_t and R_{t-1} , respectively. For the sake of convenience, we use the same convention for other symbols for the rest of this paper; i.e., if Y_t represents a frame or error signal at time t , its corresponding value at spatial location (i, j) is denoted by $Y_t(i, j)$.

In transcoding the compressed video bitstream, the output bitrate is lower than the input bitrate. As a result, the outgoing frame rate in the transcoder is usually much lower than the incoming frame rate. Hence switch SW is used to control the desired frame rate of the transcoder. Table I summarizes the operating modes of the frame-skipping transcoder.

Assume that R_{t-1} is dropped. However, R_{t-1} is required to act as the reference frame for the reconstruction of R_t such that

$$R_t(i, j) = R_{t-1}(i + u_t, j + v_t) + e_t(i, j) + \Delta_t(i, j) \quad (3)$$

where Δ_t represents the reconstruction errors of the current frame in the front-encoder due to quantization, and e_t is the residual signal between the current frame and the motion-compensated frame

$$e_t(i, j) = O_t(i, j) - R_{t-1}(i + u_t, j + v_t). \quad (4)$$

Substituting (4) into (3), we obtain an expression for R_t ,

$$R_t(i, j) = O_t(i, j) + \Delta_t(i, j). \quad (5)$$

In the transcoder, an optimized motion vector for the outgoing bitstream can be obtained by applying the motion estimation such that

$$mv_t^s = (u_t^s, v_t^s) = \arg \min_{(m, n) \in S} SAD^s(m, n) \quad (6)$$

$$SAD^s(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |R_t(i, j) - R_{t-2}^s(i + m, j + n)| \quad (7)$$

where R_{t-2}^s denotes a reconstructed frame of the previous non-skipped reference frame. The superscript “ s ” is used to denote the symbol after performing the frame-skipping transcoder. Although the optimized motion vector can be obtained by a new motion estimation, it is not desirable because of its high computational complexity. It has been a common practice to reuse incoming motion vectors. The performance is considered almost as good as a new full-scale motion estimation, and the scheme was assumed in many transcoder architectures [9], [10]. Let us represent the new motion vector as (u_t^s, v_t^s) . Hence, the reconstructed pixel in the current frame after the end-decoder is

$$R_t^s(i, j) = R_{t-2}^s(i + u_t^s, j + v_t^s) + e_t^s(i, j) + \Delta_t^s(i, j) \quad (8)$$

where $e_t^s(i, j) = R_t(i, j) - R_{t-2}^s(i + u_t^s, j + v_t^s)$ and Δ_t^s represents the requantization errors due to the re-encoding in the transcoder, then,

$$R_t^s(i, j) = R_t(i, j) + \Delta_t^s(i, j). \quad (9)$$

This equation implies that the reconstructed quality of the non-skipped frame deviates from the input sequence to the transcoder. The effect of re-encoding errors is depicted in Fig. 2 where the “Salesman” sequence was transcoded at half of the incoming frame rate. This figure shows that re-encoding errors lead to a drop in the picture quality of about 3.5 dB on average, which is a significant degradation.

III. HIGH-QUALITY FRAME-SKIPPING TRANSCODER WITH DYNAMIC CONTROL SCHEME

In [21], we proposed a direct addition of the DCT coefficients in frame-skipping transcoding to avoid re-encoding errors and to reduce the complexity for macroblocks coded without motion compensation. In this paper, we present a new frame-skipping transcoding architecture which is an extension of the work of [21]. The new architecture has three new features:

- 1) a direct addition of the DCT coefficients for macroblocks without motion compensation (non-MC macroblocks);

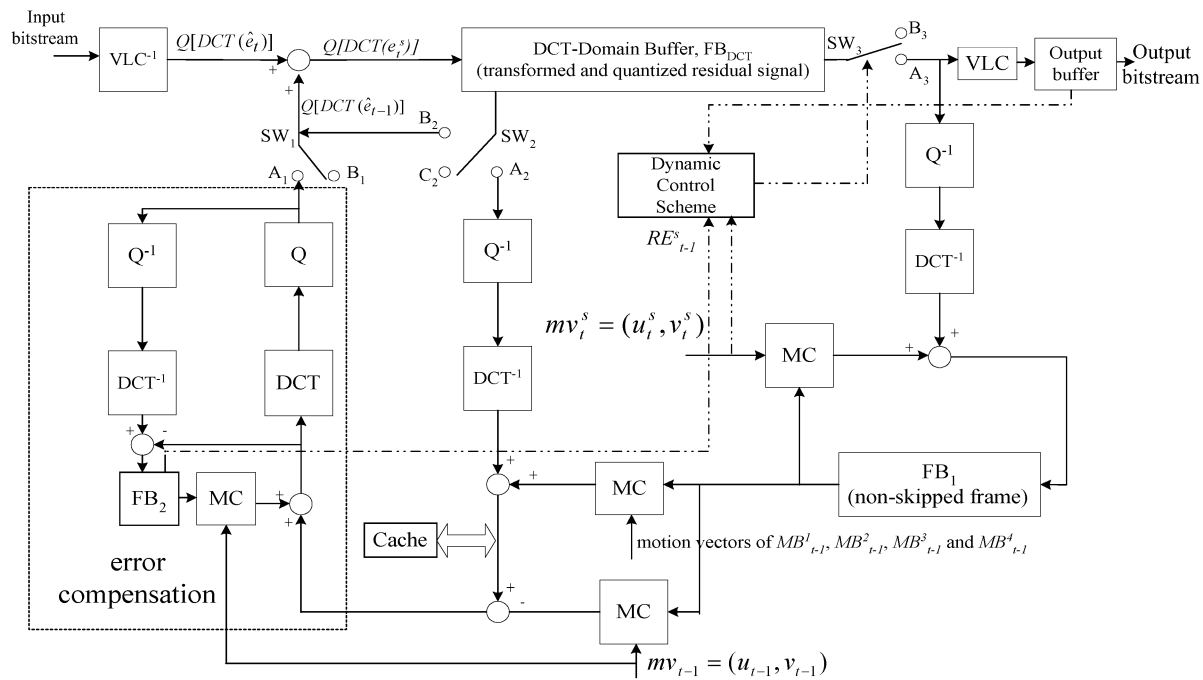


Fig. 3. Architecture proposed for frame-skipping transcoder.

TABLE II
DIFFERENT CODING MODES OF SWITCHES SW_1 AND SW_2
OF THE PROPOSED TRANSCODER

Coding mode	SW_1 Position	SW_2 Position
Non MC	B_1	B_2
MC	A_1	A_2

TABLE III
SWITCH POSITIONS FOR DIFFERENT FRAME-SKIPPING MODES
OF THE PROPOSED TRANSCODER

Frame-skipping mode	SW_3 Position
Skipped frame	B_3
Non-skipped frame	A_3

- 2) a feedback loop for error compensation within motion-compensated macroblocks (MC macroblocks);
- 3) an intelligent control scheme for a dynamic selection of the most representative and high-quality nonskipped frames.

The architecture of the proposed transcoder is shown in Fig. 3. The input bitstream is first parsed with a variable-length decoder to extract the header information, coding mode, motion vectors and quantized DCT coefficients for each macroblock. Let us define the incoming residual signal with quantization errors due to the front encoder as $\hat{e}_t = e_t + \Delta_t$ and its quantized DCT coefficients as $Q[DCT(\hat{e}_t)]$. Each macroblock is then manipulated independently. Two switches, SW_1 and SW_2 , are employed to update the DCT-domain buffer, FB_{DCT} , for the transformed and quantized residual signal depending on the coding mode originally used in the front encoder for the current macroblock being processed. The switch positions for different coding modes are shown in Table II. For non-MC macroblocks, the previous residual signal in the DCT-domain is directly fed back from FB_{DCT} to the adder, which adds the input residual signal to update FB_{DCT} . Note that all operations are performed in the DCT-domain, thus the complexity of the frame-skipping transcoder is reduced. Also, the quality degradation of the transcoder introduced by Δ_t^s is avoided. When the motion compensation mode is used, motion compensation, DCT, inverse DCT, quantization and inverse quantization modules are activated to update FB_{DCT} .

One problem with these MC macroblocks is that re-encoding errors will be generated, which introduces quality degradation in the transcoded sequence. Thus, the feedback loop in Fig. 3 tries to compensate for the re-encoding errors introduced in the previous frames. Note that switch SW_3 is used to adjust the frame rate and refresh FB_1 for the nonskipped frame. This switch is controlled by the proposed dynamic control scheme which makes use of both the motion vectors and re-encoding errors. Table III shows the frame-skipping modes of the proposed transcoder. The advantages of the DCT-domain buffer arrangement, together with the details of other methods, are described in the following subsections.

A. Direct Addition of DCT Coefficients for Non-MC Macroblocks

In Fig. 4, a situation in which one frame is dropped is illustrated. We assume that MB_t represents the current non-MC macroblock and MB_{t-1} represents the best matching macroblock with MB_t . Since MB_t is coded without motion compensation, the spatial position of MB_{t-1} is the same as that of MB_t , and MB_{t-2} represents the best matching macroblock with MB_{t-1} . Since R_{t-1} is dropped, for MB_t , we need to compute motion vector (u_t^s, v_t^s) and prediction errors in the DCT-domain, $Q[DCT(e_t^s)]$, by using R_{t-2} as a reference. Since the motion vector in MB_t is zero, then

$$(u_t^s, v_t^s) = (u_{t-1}, v_{t-1}). \quad (10)$$

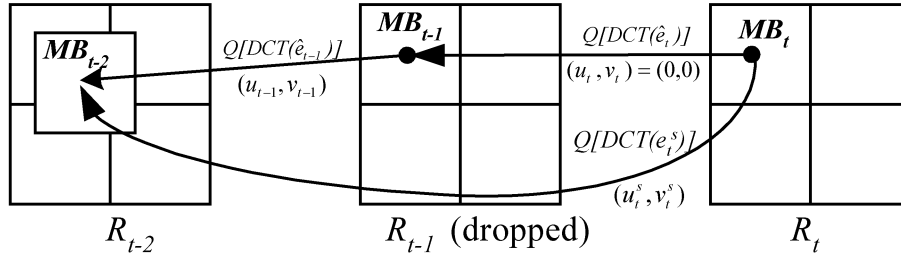


Fig. 4. Residual signal recomputation of frame skipping for non-MC macroblocks.

Re-encoding can lead to additional errors, but this can be avoided if $Q[DCT(e_t^s)]$ is computed in the DCT-domain. In Fig. 4, pixels in MB_t can be reconstructed by performing the inverse quantization and inverse DCT of $Q[DCT(\hat{e}_t)]$ and adding this residual signal to pixels in MB_{t-1} which can be similarly reconstructed by performing the inverse quantization and inverse DCT of $Q[DCT(\hat{e}_{t-1})]$ and adding this residual signal to pixels in the corresponding MB_{t-2} . The reconstructed macroblocks MB_t and MB_{t-1} are given by

$$MB_t = MB_{t-1} + \hat{e}_t \quad (11)$$

and

$$MB_{t-1} = MB_{t-2} + \hat{e}_{t-1}. \quad (12)$$

Using (11) and (12), the prediction errors between the current non-MC macroblock and its corresponding reference macroblock in R_{t-2} , $e_t^s = MB_t - MB_{t-2}$, can be written as

$$e_t^s = \hat{e}_t + \hat{e}_{t-1}. \quad (13)$$

By applying the DCT for e_t^s and taking into account the linearity of DCT, we obtain the expression of e_t^s in the DCT-domain

$$DCT(e_t^s) = DCT(\hat{e}_t) + DCT(\hat{e}_{t-1}). \quad (14)$$

Then the newly quantized DCT coefficients of prediction errors are given by

$$Q[DCT(e_t^s)] = Q[DCT(\hat{e}_t) + DCT(\hat{e}_{t-1})]. \quad (15)$$

Note that, in general, quantization is not a linear operation because of the integer truncation. However, $DCT(\hat{e}_t)$ and $DCT(\hat{e}_{t-1})$ are divisible by the quantizer step-size. Thus, we obtain the final expression of the prediction errors in the quantized DCT-domain by using R_{t-2} as a reference

$$Q[DCT(e_t^s)] = Q[DCT(\hat{e}_t)] + Q[DCT(\hat{e}_{t-1})]. \quad (16)$$

Equation (16) implies that the newly quantized DCT coefficient $Q[DCT(e_t^s)]$ can be computed in the DCT-domain by adding directly the quantized DCT coefficients between the data in the DCT-domain buffer, FB_{DCT} , and the incoming DCT coefficients, whilst the updated DCT coefficients are stored in FB_{DCT} , as depicted in Fig. 3, when switches SW_1 and SW_2 are connected to B_1 and B_2 , respectively. Since it is not necessary to perform motion compensation, DCT, quantization, inverse DCT and inverse quantization, complexity is reduced. Furthermore, since requantization is not necessary

TABLE IV
PERCENTAGE OF NON-MC MACROBLOCK FOR VARIOUS SEQUENCES

Salesman	Foreman	Carphone	Table Tennis	Football
95%	65%	69%	72%	53%

for non-MC macroblocks, re-encoding errors Δ_t^s mentioned in (9) are also avoided.

For a real-world image sequence, the block motion field is usually gentle, smooth, and varies slowly. As a consequence, the distribution of motion vector is center-biased [18]–[20], as demonstrated by the typical examples as shown in Table IV which shows the distribution of the coding modes for various sequences including “Salesman,” “Foreman,” “Carphone,” “Table Tennis,” and “Football.” These sequences have been selected to emphasize different amount of motion activities. It is clear that over 70% and 50% of the macroblocks are coded without motion compensation for sequences containing low and high amount of motion activities, respectively. By using a direct addition of the DCT coefficients in the frame-skipping transcoder, the sequence containing more non-MC macroblocks can reduce the computational complexity and re-encoding errors more significantly.

B. DCT-Domain Buffer Updating for MC Macroblocks With Error Compensation

For MC macroblocks, direct addition cannot be employed since MB_{t-1} is not on a macroblock boundary, as depicted in Fig. 5. In other words, $Q[DCT(\hat{e}_{t-1})]$ is not available from the incoming bitstream. Fig. 5 also shows that MB_{t-1} is formed by using parts of four segments which come from its four neighboring macroblocks. Let us define these four neighboring macroblocks as MB_{t-1}^1 , MB_{t-1}^2 , MB_{t-1}^3 and MB_{t-1}^4 . It is possible to use the incoming quantized DCT coefficients of MB_{t-1}^1 , MB_{t-1}^2 , MB_{t-1}^3 , and MB_{t-1}^4 , to come up with $Q[DCT(\hat{e}_{t-1})]$. First, inverse quantization and inverse DCT of the quantized DCT coefficients of MB_{t-1}^1 , MB_{t-1}^2 , MB_{t-1}^3 and MB_{t-1}^4 are performed to obtain their corresponding prediction errors in the pixel-domain. Note that each macroblock is composed of four 8×8 blocks in video coding standards [1]–[5], and the DCT and quantization operations are performed on units of 8×8 blocks. When processing MB_{t-1}^1 , MB_{t-1}^2 , MB_{t-1}^3 , and MB_{t-1}^4 , only their corresponding 8×8 blocks, which have pixels overlapping with MB_{t-1} , are subject to the inverse DCT computation. In addition to only processing blocks that have pixels overlapping with MB_{t-1} , each 8×8 block is to be inverse transformed

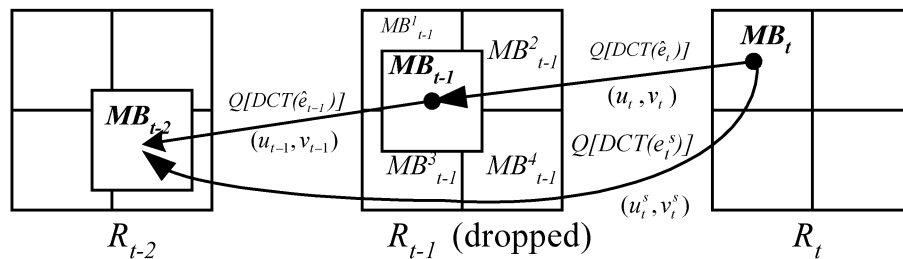


Fig. 5. Residual signal re-computation of frame-skipping for MC macroblocks.

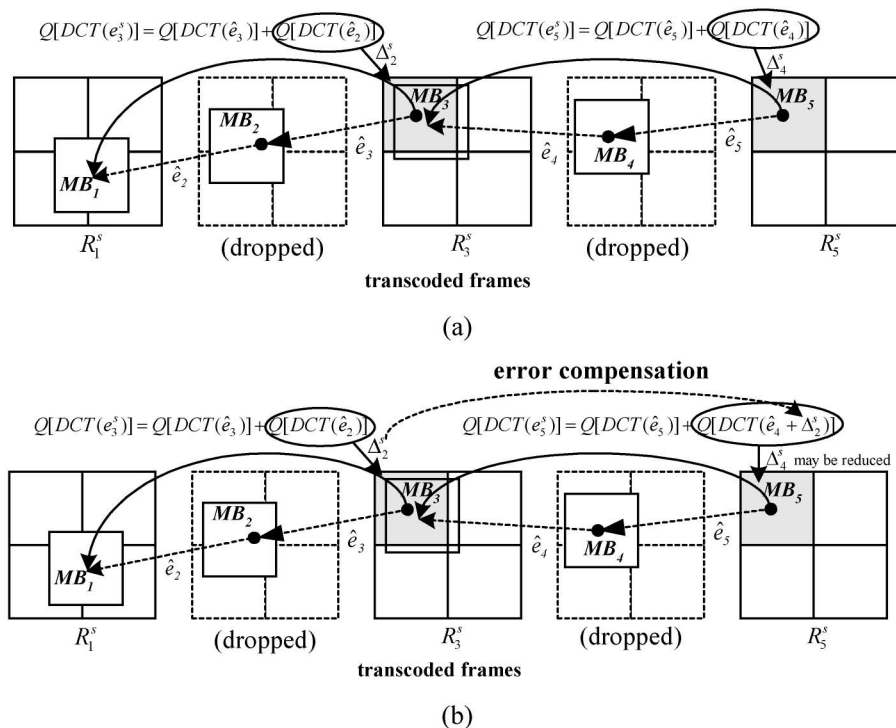


Fig. 6. Effect of re-encoding error (a) without error compensation, and (b) with error compensation.

partially. The two-dimensional (2-D) inverse DCTs are implemented using one-dimensional (1-D) row transforms followed by 1-D column transforms. All eight rows for each required block receive a 1-D inverse DCT, but only the columns that have pixels overlapping to MB_{t-1} are subjected to a 1-D inverse DCT. In most cases, the approach significantly reduces the required number of column inverse DCT.

Each segment of the reconstructed pixels in MB_{t-1} can be obtained by adding its prediction errors to its motion-compensated segment of the previous nonskipped frame stored in FB_1 , as shown in Fig. 3. After all pixels in MB_{t-1} are reconstructed, we need to find the prediction errors, \hat{e}_{t-1} . Actually, \hat{e}_{t-1} is equal to the reconstructed pixel in MB_{t-1} subtracted from its corresponding MC macroblock from the previous nonskipped frame stored in FB_1 , denoted as MB_{t-2} in Fig. 5. In order to locate MB_{t-2} , we need to find a motion vector of MB_{t-1} . Again, MB_{t-1} is not on a macroblock boundary; it is possible to use the bilinear interpolation from the motion vectors mv_{t-1, MB_1} , mv_{t-1, MB_2} , mv_{t-1, MB_3} , and mv_{t-1, MB_4} which are the four neighboring macroblocks, MB_{t-1}^1 , MB_{t-1}^2 , MB_{t-1}^3 and MB_{t-1}^4 , of MB_{t-1} to come up with an approximation of mv_{t-1} [11]. However, the bilinear interpolation

of motion vectors leads to inaccuracy in the resultant motion vector because the area covered by the four macroblocks may be too divergent and too large to be described by a single motion vector [9], [10]. Thus, the forward dominant vector selection (FDVS) method is used [9], [10] to select one dominant motion vector from four neighboring macroblocks. A dominant motion vector is defined as the motion vector carried by a dominant macroblock. The dominant macroblock is the macroblock that has the largest overlapped segment with MB_{t-1} .

Hence, \hat{e}_{t-1} can be computed and it is transformed and quantized to $Q[DCT(\hat{e}_{t-1})]$. In Fig. 5, the newly quantized DCT coefficient $Q[DCT(e_t^s)]$ of a MC macroblock can then be computed by adding $Q[DCT(\hat{e}_{t-1})]$ to the incoming $Q[DCT(\hat{e}_t)]$ and it is quite similar to that of the non-MC macroblock as mentioned in (16) except the formation of $Q[DCT(\hat{e}_{t-1})]$. For non-MC macroblocks, $Q[DCT(\hat{e}_{t-1})]$ is available from the incoming bitstreams. Conversely, requantization is performed for the formation of $Q[DCT(\hat{e}_{t-1})]$ in MC macroblocks, which will introduce additional re-encoding errors Δ_{t-1}^s such that the reconstructed frame after the end-encoder is

$$R_t^s = R_t + \Delta_{t-1}^s. \quad (17)$$

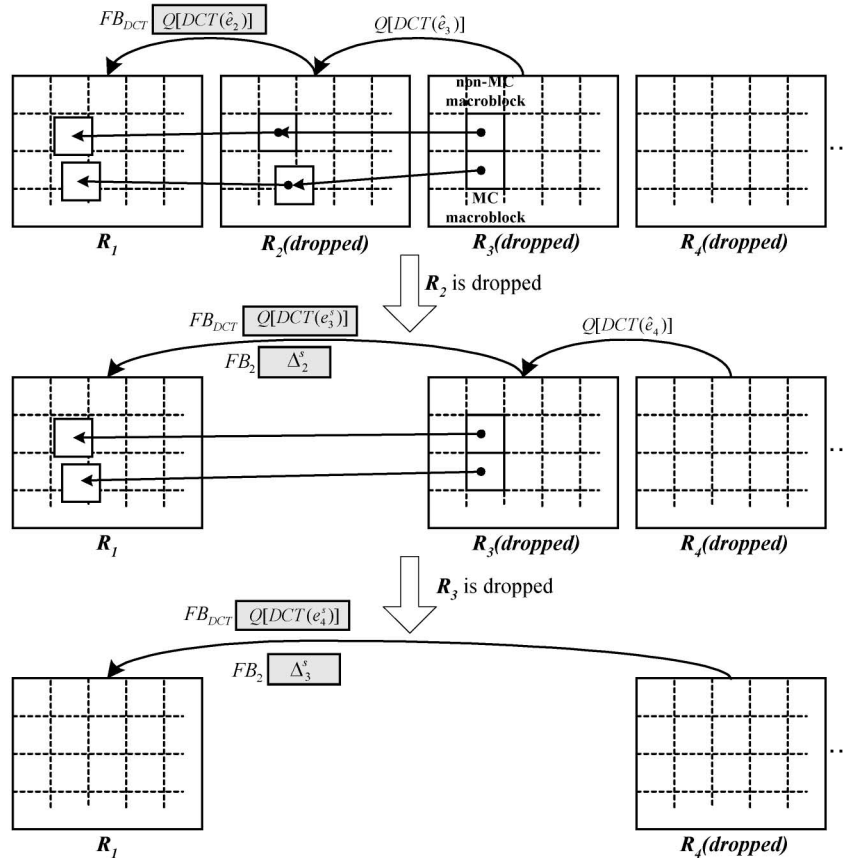


Fig. 7. Multiple frame skipping of the proposed transcoder.

Note that, as compared with Δ_t^s in (9), Δ_{t-1}^s is the re-encoding due to frame $t-1$ instead of frame t . Both of these errors will degrade the quality of the reconstructed frame. Since each nonskipped P-frame is used as a reference frame for the following nonskipped P-frame, quality degradation propagates to later frames in a cumulative manner. If the accumulated magnitude of re-encoding errors is large, it means that the quality of the transcoded sequence is degraded significantly. This is illustrated in Fig. 6(a) which shows how re-encoding can lead to accumulated errors. According to the figure, Δ_2^s is introduced for the formation of $Q[DCT(\hat{e}_2)]$ and such errors have the effect of degrading the reconstructing quality of MB_3 . When pixels in MB_3 are used as a reference for the subsequent nonskipped frame, for example, R_5 in Fig. 6(a), Δ_2^s will further affect the formation of $Q[DCT(\hat{e}_4)]$ and finally this error is accumulated in MB_5 . These accumulated errors become significant in the sequence containing a large amount of MC macroblocks.

With the possibility of having re-encoding errors in MC macroblocks, it is obviously important to develop techniques to minimize the visual degradation caused by this phenomenon. Thus, a feedback loop is suggested as shown in Fig. 3 to compensate for the re-encoding errors introduced in the previous frames. The forward and inverse DCT and quantization pairs in the feedback loop are mainly responsible for minimizing re-encoding errors. For these MC macroblocks, the quantized DCT coefficients are inversely quantized. An inverse DCT is then performed to form \hat{e}_{t-1} with a re-encoding error, which subtracts

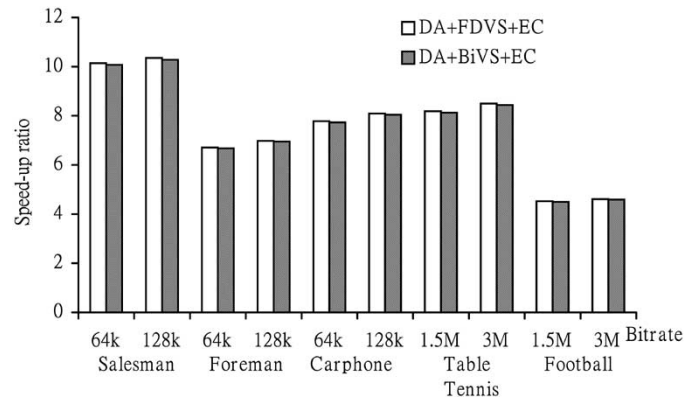


Fig. 8. Speed-up ratio of various transcoders as compared with CPDT+FDVS, where the frame rate of the incoming bitstream was 30 frames/s and then transcoded to 10 frames/s. The front encoder for encoding "Salesman," "Foreman," and "Carphone" was H.263 TMN8 [26]; while MPEG2 TM5 [27] was used to encode "Table Tennis" and "Football."

the original \hat{e}_{t-1} to generate the re-encoding error, Δ_{t-1}^s . The re-encoding error is stored in FB_2 can be written as

$$\Delta_{t-1}^s = DCT^{-1} \left(\left\lfloor \frac{DCT(\hat{e}_{t-1})}{q} \right\rfloor \times q \right) - \hat{e}_{t-1} \quad (18)$$

where q is the quantization step-size and the floor function, $\lfloor a \rfloor$, extracts the integer part of the given argument a .

TABLE V
SIMULATION CONDITIONS

	Conventional pixel-domain transcoder		Proposed transcoders			
	CPDT + FDVS	CPDT + FDVS + FSC _t ^s (MA _t ^s)	DA + FDVS	DA + FDVS + EC	DA + BiVS + EC	DA + FDVS + EC + FSC _t ^s (MA _t ^s , RE _{t-1} ^s)
Motion vector composition method	FDVS	FDVS	FDVS	FDVS	BiVS	FDVS
Direct addition (DA)	—	—	ON	ON	ON	ON
Error compensation (EC)	—	—	OFF	ON	ON	ON
Dynamic control scheme	OFF	FSC _t ^s (MA _t ^s)	OFF	OFF	OFF	FSC _t ^s (MA _t ^s , RE _{t-1} ^s)

Since the motion vectors are highly correlated in the successive frames [22], [23], it is observed that the spatial positions of MC macroblocks in certain frames are very close to the spatial positions of MC macroblocks in its subsequent frames. Thus, re-encoding errors stored in FB_2 are added to the prediction errors of MC macroblocks in the following P-frame to compensate for the re-encoding errors. For example, as shown in Fig. 6(b), Δ_2^s is added to \hat{e}_4 such that it is able to compensate for the re-encoding errors for the formation of $Q[DCT(\hat{e}_4)]$. Note that the feedback loop for error compensation cannot ensure the elimination of all the re-encoding errors generated by MC macroblocks and there still exists a certain amount of re-encoding errors after frame-skipping transcoding. However, these re-encoding errors are continuously accumulated in FB_2 such that most of them can be compensated for in the subsequent frames if the spatial positions of MC macroblocks between successive frames are highly correlated.

In order to reduce the implementation complexity of the MC macroblock, a cache subsystem is added to the proposed transcoder, as depicted in Fig. 3. Since motion compensation of multiple macroblocks may require the same pixel data, a cache subsystem is implemented to reduce the redundant inverse quantization, inverse DCT and motion compensation computations. We have found that the arrangement is significant since the frequency of caching hits is high. This is due to the fact that the locality of motion often exists within each frame.

C. Buffer Arrangements for Multiple Frame Skipping

In Fig. 3, it can be seen that three frame buffers were arranged in deriving the proposed architecture of frame-skipping transcoder. FB_1 is used to store the previous nonskipped frame. Since the main feature of the proposed transcoder is to operate the frame skipping in the DCT domain, the quantized DCT coefficients are updated in FB_{DCT} . To enhance the performance of the proposed transcoder, FB_2 is employed to store re-encoding errors to compensate for the accumulated errors. When multiple frames are dropped, the proposed frame-skipping transcoder can be processed in the forward order, thus eliminating the requirement for multiple buffers in FB_{DCT} and FB_2 which could be required to store the quantized DCT coefficients and re-encoding errors of all skipped frames, respectively. Fig. 7 shows a scenario when multiple frames are dropped. Assume that R_1 is the first nonskipped frame. At the beginning, the contents of FB_{DCT} and FB_2 are initialized to zero. When R_2 is input to the transcoder, we directly store the incoming $Q[DCT(\hat{e}_2)]$ in FB_{DCT} since

there is no skipped frame between R_1 and R_2 . The stored coefficients of FB_{DCT} are used to update the DCT coefficients of the next skipped frame. This means that when R_2 is dropped, the proposed transcoder updates the DCT coefficients of prediction errors for each macroblock according to its coding mode. From (16), for non-MC macroblocks, $Q[DCT(e_3^s)]$ is obtained by a direct addition of $Q[DCT(\hat{e}_2)]$ in FB_{DCT} to the incoming $Q[DCT(\hat{e}_3)]$ of R_3 . On the other hand, it is necessary to perform the re-encoding of non-MC macroblocks in order to compute $Q[DCT(\hat{e}_2)]$ which is added to the incoming $Q[DCT(\hat{e}_3)]$ for the formation of $Q[DCT(e_3^s)]$, as mentioned in Section III-B, FB_{DCT} is then updated with the new $Q[DCT(e_3^s)]$. Simultaneously, re-encoding errors Δ_2^s are stored in FB_2 . For the next incoming frame R_4 , $Q[DCT(e_4^s)]$ with error compensation can be iteratively computed as

$$Q[DCT(e_4^s)] = Q[DCT(\hat{e}_4)] + Q[DCT(e_3^s + \Delta_2^s)]. \quad (19)$$

Again, the re-encoding of $e_3^s + \Delta_2^s$ of non-MC macroblocks will generate accumulated errors Δ_3^s which are stored in FB_2 . This iterative process has the advantage that only one pair of FB_{DCT} and FB_2 is needed for all skipped frames. The flexibility of multiple frame-skipping provides the fundamental framework for dynamic frame-skipping as described in the following.

D. Dynamic Frame-Skipping Transcoder

Our proposed frame-skipping transcoder also develops a strategy for determining the length of the skipped frame such that it can reduce the quality degradation as well as minimize the motion jerkiness perceived by human beings. Traditionally, a motion vector is used to serve as a good indicator for dynamic frame skipping [11]. When multiple frames are dropped in the frame-skipping transcoder, re-encoding errors in MC macroblocks cannot be avoided entirely even though a feedback loop is applied to compensate for the accumulated errors, as mentioned in the previous section. It is observed that human eyes are sensitive to this type of quality degradation. Thus, it is necessary to regulate the frame rate of the transcoder by taking into account the effect of re-encoding. The goal of the proposed dynamic frame-rate control scheme is to minimize the re-encoding errors as well as to preserve motion smoothness. To obtain a quantitative measure for frame-skipping, let us define a frame-skipping metric, $FSC_t^s(MA_t^s, RE_{t-1}^s)$, which is a function of the accumulated magnitude of the motion vectors

TABLE VI
AVERAGE PSNR OF THE PROPOSED TRANSCODERS WITHOUT FRAME-RATE CONTROL, WHERE THE FRAME RATE OF THE INCOMING BITSTREAM WAS 30 FRAMES/S. THE FRONT ENCODER FOR ENCODING "SALESMAN," "FOREMAN," AND "CARPHONE" WAS H.263 TM8 [26]; WHILE MPEG2 TM5 [27] WAS USED TO ENCODE "TABLE TENNIS," AND "FOOTBALL"

Sequences	Input bitrate	CPDT+FDVS			DA+FDVS			DA+BIVS+EC			DA+FDVS+EC		
		MC region	Non-MC region	All region	MC Region	Non-MC region	All region	MC region	Non-MC region	All region	MC region	Non-MC region	All region
Output sequences are transcoded to 15 frames/s													
Salesman (176x144)	64k	30.76	33.54	33.20	31.69	35.77	35.33	31.56	35.82	35.37	31.99	35.89	35.42
	128k	34.41	36.91	36.71	34.86	39.14	38.62	34.60	39.22	38.69	35.03	39.30	38.78
Foreman (176x144)	64k	29.72	31.06	30.87	30.95	33.16	32.28	31.05	33.32	32.38	32.25	33.48	32.80
	128k	33.25	34.98	34.64	34.78	37.03	36.05	34.92	37.23	36.22	36.10	37.35	36.60
Carphone (176x144)	64k	30.86	32.59	32.28	31.29	34.47	33.71	31.38	34.62	33.86	32.36	34.77	34.14
	128k	33.40	35.25	34.92	34.89	37.12	36.25	35.14	37.28	36.37	36.01	37.39	36.64
Table Tennis (352x240)	1.5M	30.89	31.57	31.41	31.26	34.59	34.00	31.39	34.74	34.16	32.07	34.84	34.19
	3M	33.92	34.60	34.44	34.74	37.19	36.82	34.89	37.34	36.95	35.68	37.45	37.04
Football (352x240)	1.5M	29.69	30.73	30.03	30.89	32.76	32.04	31.05	32.94	32.18	32.16	33.08	32.64
	3M	33.60	34.34	33.87	35.19	36.41	36.02	35.46	36.62	36.25	36.57	36.78	36.68
Output sequences are transcoded to 10 frames/s													
Salesman (176x144)	64k	30.64	33.72	33.28	31.48	35.71	35.10	31.35	35.74	35.12	31.83	35.84	35.31
	128k	34.42	36.99	36.77	34.76	39.15	38.53	34.59	39.18	38.54	35.01	39.26	38.76
Foreman (176x144)	64k	29.58	30.92	30.73	30.21	32.58	31.69	30.26	33.09	32.24	31.61	33.29	32.70
	128k	33.16	34.92	34.57	33.68	36.08	35.15	33.77	36.75	35.81	35.03	36.90	36.42
Carphone (176x144)	64k	30.87	32.49	32.25	31.06	33.89	32.93	31.14	34.40	33.49	32.17	34.59	33.96
	128k	33.41	35.16	34.90	33.76	36.03	35.32	34.31	36.82	36.12	35.21	36.94	36.45
Table Tennis (352x240)	1.5M	30.68	31.47	31.36	31.05	34.24	33.35	31.17	34.51	33.57	31.80	34.67	33.87
	3M	33.76	34.48	34.28	34.19	36.29	35.70	33.97	36.89	36.07	34.83	37.06	36.44
Football (352x240)	1.5M	29.26	30.62	29.96	30.22	32.16	31.23	30.27	32.69	31.53	31.58	32.89	32.26
	3M	33.32	34.25	33.80	34.16	35.74	34.98	34.28	36.08	35.22	35.49	36.31	35.92
Output sequences are transcoded to 7.5 frames/s													
Salesman (176x144)	64k	29.90	33.71	33.20	31.33	35.56	34.94	31.19	35.60	35.07	31.72	35.73	35.22
	128k	34.36	36.92	36.69	34.57	38.98	38.36	34.42	39.05	38.49	34.95	39.16	38.74
Foreman (176x144)	64k	29.32	30.64	30.45	29.63	32.38	31.28	29.65	33.03	32.02	31.05	33.26	32.65
	128k	32.93	34.82	34.06	33.06	35.52	34.45	33.11	36.65	35.56	34.41	36.86	36.18
Carphone (176x144)	64k	30.89	32.40	32.21	31.98	33.49	32.55	31.09	34.20	33.28	32.04	34.37	33.74
	128k	33.40	35.09	34.87	33.64	35.76	35.05	33.98	36.75	35.87	35.08	36.89	36.21
Table Tennis (352x240)	1.5M	30.66	31.37	31.17	31.07	33.92	33.12	31.11	34.26	33.38	31.65	34.42	33.65
	3M	33.73	34.40	34.21	34.10	36.05	35.51	33.71	32.61	32.92	34.71	36.95	36.32
Football (352x240)	1.5M	28.97	30.33	29.68	29.62	31.88	30.81	29.70	32.60	31.21	31.05	32.85	31.99
	3M	33.09	34.12	33.63	33.58	35.24	34.44	33.56	35.95	34.81	34.86	36.26	35.59

and re-encoding errors due to transcoding for the macroblocks of the current frame. The metric is given by

$$FSC_t^s(MA_t^s, RE_{t-1}^s) = \frac{\sum_{m=1}^M (MA_t^s)_m}{\sum_{m=1}^M (RE_{t-1}^s)_m} \quad (20)$$

where M is the total number of macroblocks in the current frame. Re-encoding errors after error compensation $(RE_{t-1}^s)_m$ are obtained by adding all requantization errors for the m th macroblock which can be written as

$$(RE_t^s)_m = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \Delta_{t-1}^s(i, j) \quad (21)$$

where N is the size of macroblock and the corresponding motion activity $(MA_t^s)_m$ is given by

$$(MA_t^s)_m = |(u_t^s)_m| + |(v_t^s)_m| \quad (22)$$

where $(u_t^s)_m$ and $(v_t^s)_m$ are the horizontal and vertical components of the motion vector of the m th macroblock which uses the previous nonskipped frame as its reference.

If the value of $FSC_t^s(MA_t^s, RE_{t-1}^s)$ following a nonskipped frame exceeds a predefined threshold, T_{FSC} , this incoming frame should be kept. $(MA_t^s)_m$ in (22) is used to detect the activity level of the m th macroblock. If $\sum_{m=1}^M (MA_t^s)_m$ has a significant value, this implies that the incoming frame contains a certain amount of motion activities. It is reasonable that the frame-rate control scheme be used to keep this frame since the previous nonskipped frame is not sufficient to represent the current frame. As a consequence, it is much better that the incoming frame be refreshed. However, we cannot guarantee the quality of the reconstructed frame due to re-encoding errors in the transcoder in cases where only the motion activity is used. Since the quality of the selected frame directly affects the motion smoothness of the transcoded sequence, it is usually beneficial to maintain the selected frame at a good reconstruction quality. The conventional algorithm fails to meet this objective. Thus, $\sum_{m=1}^M (RE_t^s)_m$ in (20) is used to measure

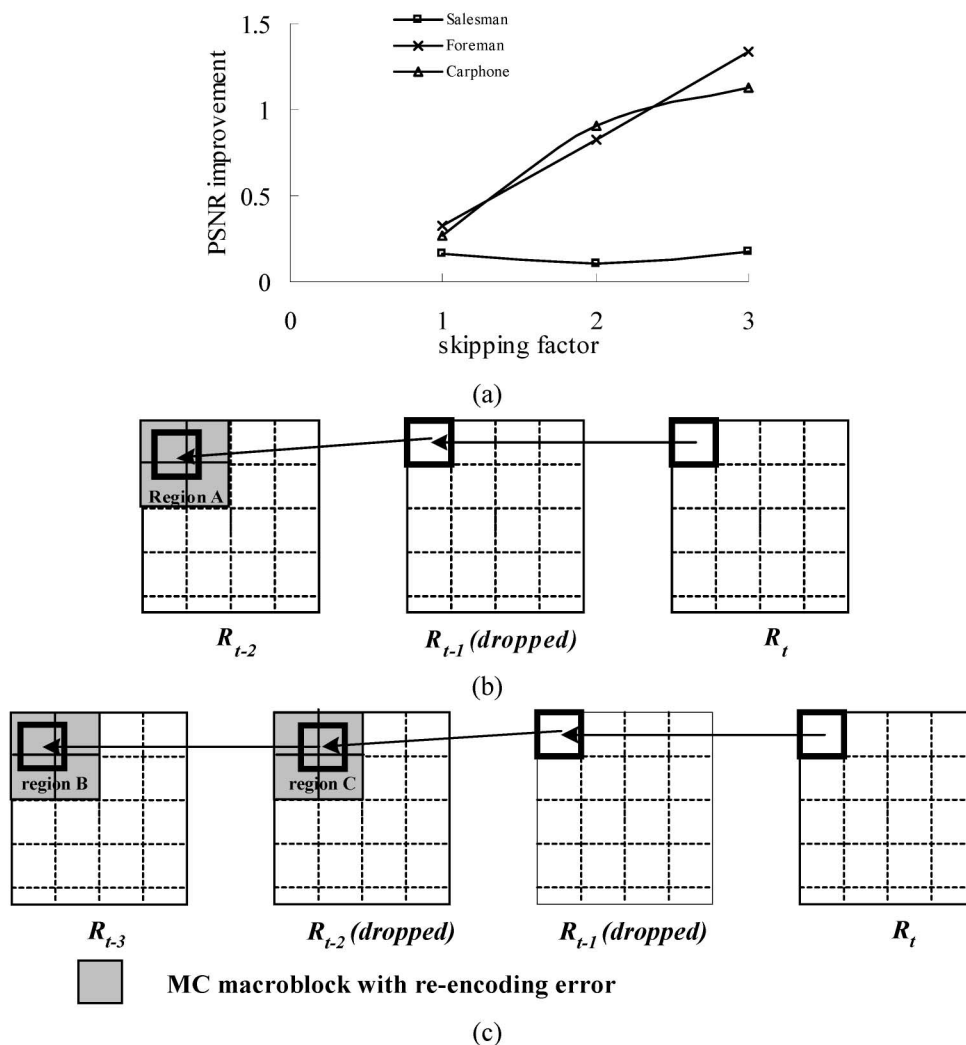


Fig. 9. Effect of error compensation in multiple frame skipping. (a) Average PSNR improvement of the non-MC macroblock against skipping factor for "Salesman," "Foreman," and "Carphone" sequences encoded at 128 Kb/s. (b) Accumulated error in single frame skipping. (c) Accumulated error in multiple frame skipping.

re-encoding errors in the incoming frame. Also, a larger value of $\sum_{m=1}^M (RE_t^s)_m$ implies more re-encoding errors, and it will reduce the value of $FSC_t^s(MA_t^s, RE_{t-1}^s)$ such that the incoming frame is not kept even though it contains a certain amount of motion activities. Note that the threshold, T_{FSC} , is simply set according to the target frame rate of the transcoder, f_T . The feedback signal from the output buffer to the dynamic control scheme in Fig. 3 is used to stabilize the outgoing frame rate of the transcoder, f_o , by adjusting the value of T_{FSC} dynamically. Initially, T_{FSC} is set to its initial value T_{init} . T_{FSC} can then be updated as follows:

- if $f_o > f_T$, increase T_{FSC} by T_{step} ;
- if $f_o < f_T$, decrease T_{FSC} by T_{step} ;
- otherwise, keep the current value of T_{FSC} ;

where T_{step} in above are the step size for adjusting T_{FSC} .

IV. SIMULATION RESULTS

Simulations have been performed to evaluate the overall efficiency of various frame-skipping transcoders. Two sets

of experiments were carried out. First, in the front encoder, the first frame was encoded as intraframe (I-frame), and the remaining frames were encoded as interframes (P-frames). In the first simulation, bi-directional frames (B-frames) were not considered. Second, we demonstrate the performance of the proposed transcoder when the incoming bitstreams were encoded with B-frames. In both simulations, picture-coding modes were preserved during transcoding.

A. Performance of the Proposed Techniques on Frame-Skipping Transcoder

The first set of experiments aims at evaluating the performances of the proposed techniques including the direct addition for non-MC macroblocks (DA), the error compensation for MC macroblocks (EC) and the dynamic selection of nonskipped frames by employing the incoming motion vectors and the re-encoding error ($FSC_t^s(MA_t^s, RE_{t-1}^s)$) when applied to the frame-skipping transcoder. Different front encoders were employed to encode two sets of video sequences with different spatial resolutions and motion characteristics. "Salesman," "Foreman," and "Carphone" are typical videophone sequences

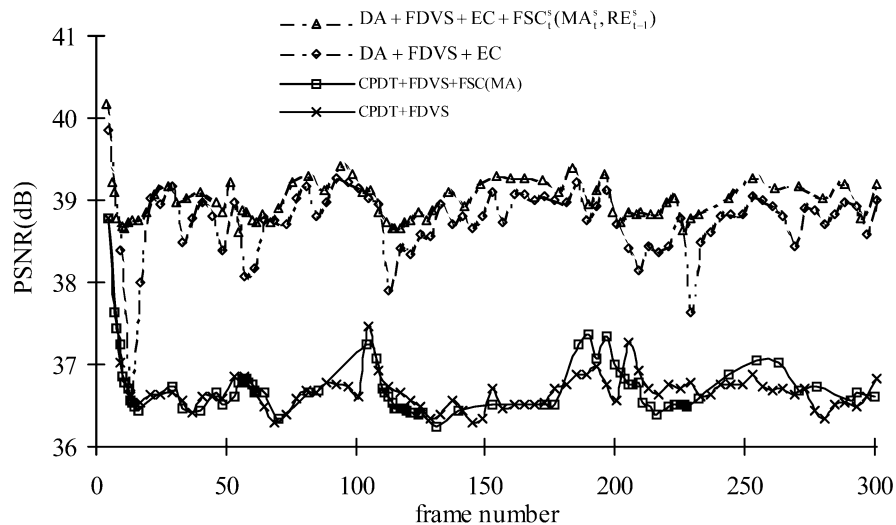


Fig. 10. PSNR of the proposed dynamic frame-skipping transcoder of “Salesman” sequence encoded at 128 Kb/s with 30 frames/s, and then transcoded to 32 Kb/s with about 7.5 frames/s.

in QCIF (176×144) format, which are used to show the performance of the proposed frame-skipping transcoder in multipoint video conferencing [7]. H.263 is currently the best standard for practical video conferencing and its range of target bitrates is about 10–2048 Kb/s. Since H.263 could be superior to H.261 at any bitrate [24], [25], an H.263 TMN8 front encoder [26] was employed to encode “Salesman,” “Foreman,” and “Carphone” at different bitrates (64 Kb/s and 128 Kb/s). On the other hand, “Tennis” and “Football” in SIF (352×240) containing high motion activities were encoded by a MPEG2 TM5 front encoder [27] at different bitrates (1.5 Mb/s and 3 Mb/s), but only P-frames were generated. For all testing sequences, the frame-rate of the incoming bitstream was 30 frames/s.

To verify the performances of the proposed techniques, extensive simulations were carried out. Results of the simulations are used to compare the performance of a reference transcoder which is a conventional pixel-domain transcoder (CPDT) by employing FDVS to compose an outgoing motion vector from the incoming motion vectors of the skipped frames [9], [10], named as CPDT+FDVS. Table V shows the simulation conditions for different transcoders examined. Note that BiVS represents the bilinear interpolation vector selection for composing an outgoing motion vector [11]. The detailed comparisons of the average PSNR between CPDT+FDVS and the proposed transcoders including DA+FDVS, DA+BiVS+EC, and DA+FDVS+EC are tabulated in Table VI in which the frames are temporally dropped by a factor of 1, 2, and 3. We show that both DA+FDVS, DA+BiVS+EC and DA+FDVS+EC outperform CPDT+FDVS in all cases. The results are more significant for the non-MC macroblock because a direct addition of the DCT coefficients should not introduce any re-encoding errors. Also, Fig. 8 shows that the proposed transcoders have a speed-up of about 4.5–10 times faster than that of CPDT+FDVS. This is because the probability of the non-MC macroblock happens more frequently in typical sequences, and we can achieve significant computational savings while maintaining good video quality on these non-MC macroblocks. In addition, the cache system in

the proposed transcoders can reduce the computational burden of re-encoding the MC macroblocks.

Table VI and Fig. 8 also compare the average PSNR and complexities of two transcoders using different approaches for composing outgoing motion vectors: DA+FDVS+EC and DA+BiVS+EC. As shown in Table VI, DA+FDVS+EC is consistently better than DA+BiVS+EC at various outgoing frame rates. It is significant to note that the inaccuracy of the resultant motion vector of BiVS affects the average PSNR of the MC macroblock. Also, the complexity of DA+BiVS+EC is slightly higher than that of DA+FDVS+EC as shown in Fig. 8. Therefore, FDVS is more suitable for frame-skipping transcoders.

In Table VI, it can be seen that DA+FDVS+EC has a PSNR improvement over DA+FDVS. This result is expected since the feedback loop of DA+FDVS+EC is enabled which can reduce the re-encoding errors of MC macroblocks. In other words, the average PSNR of the MC macroblock of DA+FDVS+EC is better than that of DA+FDVS. The advantage of error compensation is significant for sequences containing high motion activities. In the “Salesman” sequence, the average PSNR of DA+FDVS+EC is almost the same as that of DA+FDVS. However, DA+FDVS+EC significantly outperforms DA+FDVS for all “Foreman,” “Carphone,” “Table Tennis,” and “Football” sequences which contain certain amount of motion activities. There is also about 0.3 dB improvement of the non-MC macroblock in high moving sequences in which they are transcoded into half of the incoming frame rate. Let us use Fig. 9(b) to give a clearer account of this phenomenon. For DA+FDVS, pixels in region A have re-encoding errors since these pixels are located at the MC macroblocks of R_{t-2} . These errors can be accumulated to the non-MC macroblock in R_t . The PSNR improvement of the non-MC macroblock of DA+FDVS+EC is due to the contribution from the feedback loop of error compensation in which requantization errors are stored in FB_2 and is fed back to the quantizer to compensate for the requantization errors introduced in the previous frame. In Fig. 9(a), we plot the

TABLE VII
AVERAGE PSNR AND SPEED-UP RATIO OF VARIOUS DYNAMIC TRANSCODERS AS COMPARED WITH CPDT+FDVS
USING H.263 TMN8 [26] AS A FRONT ENCODER

Sequence	Method	Average PSNR	Speed-up ratio	Average encoded frames per second
Input bitrate is 64kbps with 30 frames/s				
Salesman	CPDT+FDVS	33.20	-	7.5
	CPDT + FDVS+ $FSC_t^s(MA_t^s)$	33.24	0.94	7.8
	DA + FDVS+ EC	35.22	9.95	7.5
	DA + FDVS+ EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	35.48	8.16	7.5
Foreman	CPDT+FDVS	30.45	-	7.5
	CPDT + FDVS+ $FSC_t^s(MA_t^s)$	30.48	0.96	7.4
	DA + FDVS+ EC	32.65	6.45	7.5
	DA + FDVS+ EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	32.95	4.95	7.4
Carphone	CPDT+FDVS	32.21	-	7.5
	CPDT + FDVS+ $FSC_t^s(MA_t^s)$	32.27	0.96	7.4
	DA + FDVS+ EC	33.74	7.38	7.5
	DA + FDVS+ EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	33.97	6.15	7.4
Input bitrate is 128kbps with 30 frames/s				
Salesman	CPDT+FDVS	36.69	-	7.5
	CPDT + FDVS+ $FSC_t^s(MA_t^s)$	36.72	0.94	7.8
	DA + FDVS+ EC	38.74	10.15	7.5
	DA + FDVS+ EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	38.98	8.31	7.5
Foreman	CPDT+FDVS	34.06	-	7.5
	CPDT + FDVS+ $FSC_t^s(MA_t^s)$	34.12	0.96	7.4
	DA + FDVS+ EC	36.18	6.62	7.5
	DA + FDVS+ EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	36.46	5.54	7.4
Carphone	CPDT+FDVS	34.87	-	7.5
	CPDT + FDVS+ $FSC_t^s(MA_t^s)$	34.92	0.96	7.4
	DA + FDVS+ EC	36.21	7.65	7.5
	DA + FDVS+ EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	36.49	6.42	7.4

average PSNR improvement of the non-MC macroblock for different skipping factors. For sequences containing high motion activities such as ‘‘Foreman’’ and ‘‘Carphone,’’ significant improvement in average PSNR, of about 0.85 dB and 1.25 dB, have been achieved when the frames are temporally dropped by a factor of 2 and 3, respectively. The reason for this is illustrated in Fig. 9(c) when the skipping factor is equal to 2. Pixels in both regions B and C will contribute accumulated errors in the non-MC macroblock of R_t and the effect of accumulated errors is serious. However, the technique of error compensation can reduce these accumulated errors. The results of ‘‘Table Tennis’’ and ‘‘Football’’ sequences appear to be similar to that of the above, as shown in Table VI. On the other hand, the improvement for the ‘‘Salesman’’ sequence is not remarkable, because the sequence has only low motion activities.

Even though DA+FDVS+EC can greatly improve the overall performance as compared with CPDT+FDVS, the abrupt change in PSNR is significant, as shown in Fig. 10 and it affects the motion smoothness of the transcoded sequence which probably introduces a flickering effect. Although the fluctuation of PSNR is not an exact measure of the flickering effect, it is fair to say that the flickering effect can be reduced by a smaller fluctuation of PSNR. DA+FDVS+EC is further enhanced by incorporating the proposed frame rate control scheme $FSC_t^s(MA_t^s, RE_{t-1}^s)$, named as DA+FDVS+EC+ $FSC_t^s(MA_t^s, RE_{t-1}^s)$. We have set the parameters T_{init} and T_{step} in $FSC_t^s(MA_t^s, RE_{t-1}^s)$ to 20 and 5, respectively, for the rest of the comparison. The PSNR

performance of DA+FDVS+EC+ $FSC_t^s(MA_t^s, RE_{t-1}^s)$ in transcoding various video sequences is shown in Tables VII, VIII, and Fig. 10, for which the target frame rate f_T was set to 7.5 frames/s. In Tables VII and VIII, DA+FDVS+EC+ $FSC_t^s(MA_t^s, RE_{t-1}^s)$ increases the average PSNR of the proposed transcoder by about 0.2–0.3 dB while it reduces the fluctuation of PSNR, as depicted in Fig. 10. The figure also shows that $FSC_t^s(MA_t^s, RE_{t-1}^s)$ outperforms the conventional frame-rate control scheme by using the incoming motion vectors only, $FSC_t^s(MA_t^s)$ [11]. This is because $FSC_t^s(MA_t^s, RE_{t-1}^s)$ can reduce re-encoding errors by preserving the high quality reference frame with high motion activities. These results show that $FSC_t^s(MA_t^s, RE_{t-1}^s)$ is able to strengthen our new architecture of frame-skipping transcoder and provides the decoded sequence with a smoother motion as well as better transcoded pictures.

B. Results With Bi-Directional Frames (B-Frames)

In this simulation, ‘‘Tennis,’’ ‘‘Football,’’ and ‘‘Stefan’’ sequences were encoded by a MPEG-2 TM5 front encoder with the structure ‘‘IBBPBBPBBP ...’’ as the group of pictures (GOP). The bitstreams were then transcoded into lower frame rates. In Fig. 11, frames are presented in the display order, but are numbered in the encoding order. Considering that the first input frame to the transcoder is I_o and its second one is P_1 . The dynamic transcoding of P-frames involves the process of selecting the most representative frames according to the frame-skipping metric mentioned in Section III-D ($FSC_t^s(MA_t^s, RE_{t-1}^s)$) or

TABLE VIII
AVERAGE PSNR AND SPEED-UP RATIO OF VARIOUS DYNAMIC TRANSCODERS AS COMPARED WITH CPDT+FDVS
USING MPEG2 TM5 [27] AS A FRONT ENCODER

Sequence	Method	Average PSNR	Speed-up ratio	Average encoded frames per second
Input bitrate is 1.5Mbps with 30 frames/s				
Table Tennis	CPDT+FDVS	31.17	-	7.5
	CPDT + FDVS + $FSC_t^s(MA_t^s)$	31.24	0.96	7.4
	DA + FDVS + EC	33.65	7.86	7.5
	DA + FDVS + EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	33.88	6.65	7.4
Football	CPDT+FDVS	29.68	-	7.5
	CPDT + FDVS + $FSC_t^s(MA_t^s)$	29.77	0.97	7.4
	DA + FDVS + EC	31.99	4.39	7.5
	DA + FDVS + EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	32.30	3.85	7.4
Input bitrate is 3Mbps with 30 frames/s				
Table Tennis	CPDT+FDVS	34.21	-	7.5
	CPDT + FDVS + $FSC_t^s(MA_t^s)$	34.31	0.96	7.4
	DA + FDVS + EC	36.32	8.17	7.5
	DA + FDVS + EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	36.59	6.91	7.4
Football	CPDT+FDVS	33.63	-	7.5
	CPDT + FDVS + $FSC_t^s(MA_t^s)$	33.70	0.97	7.4
	DA + FDVS + EC	35.59	4.48	7.5
	DA + FDVS + EC + $FSC_t^s(MA_t^s, RE_{t-1}^s)$	35.91	3.96	7.4

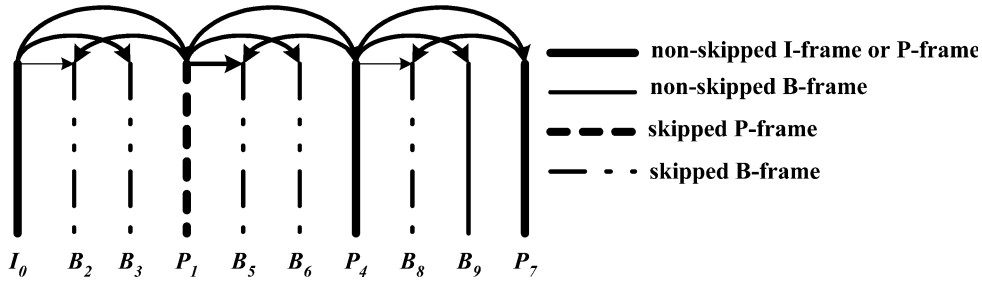


Fig. 11. Input and output frames in the display order, but are numbered in the encoding order.

the conventional scheme ($FSC_t^s(MA_t^s)$). The next incoming bidirectional frames, B_2 and B_3 , are predicted from I_0 and P_1 . Due to this dependence, the selection of B-frames, B_2 and B_3 , has to be postponed after their referenced frames, I_0 and P_1 , are transcoded. One straightforward approach to selecting a B-frame, which depends upon its referenced P-frames, is discussed as follows.

- 1) If one of its referenced frames is not selected, this B-frame is dropped. For example, as illustrated in Fig. 11, when P_1 is not selected, all frames which reference to P_1 such as B_2 , B_3 , B_5 and B_6 are dropped.
- 2) If both of its referenced frames are selected, the B-frames between their reference frames may also be significant. In such cases, for each B-frame, we can approximate its value of frame-skipping metric, by assuming this metric between the frames is uniform in a short period of time, such that the frame-skipping metric of B-frame is a scaled version of its corresponding metric of P-frame. In Fig. 11, P_4 and P_7 are selected. Then, the frame-skipping metrics of B_8 and B_9 become $(1/3)FSC_7^s$ and $(2/3)FSC_7^s$, respectively. If the scaled metric of B-frame is larger than T_{FSC} , this B-frame is selected. Otherwise, it is dropped.

In transcoding the B-frame, the quantized DCT coefficients of nonskipped B-frames can be directly obtained from the incoming bitstream because both of its reference frames are available. Besides, since a B-frame is not used as reference for further prediction, it is not necessary to update all buffers in the frame-skipping transcoder. This technique can be easily integrated into the DA+FDVS+EC+ $FSC_t^s(MA_t^s)$ and DA+FDVS+EC+ $FSC_t^s(MA_t^s, RE_{t-1}^s)$, and the results are shown in Fig. 12 and Table IX. It is seen that the proposed DA+FDVS+EC+ $FSC_t^s(MA_t^s, RE_{t-1}^s)$ has about a 2.5-dB PSNR improvement as compared to that of DA+FDVS+EC+ $FSC_t^s(MA_t^s)$. The complexity of DA+FDVS+EC+ $FSC_t^s(MA_t^s, RE_{t-1}^s)$ is also less than that of DA+FDVS+EC+ $FSC_t^s(MA_t^s)$. These demonstrate the effect of the proposed frame-skipping transcoder when the incoming bitstream contains B-frames.

V. CONCLUSIONS

In this paper, we have proposed a new architecture for a low-complexity and high quality frame-skipping transcoder. Its low complexity is achieved by: 1) a direct addition of the DCT coefficients for macroblocks coded without motion compensation to deactivate most of the complex modules of the transcoder and

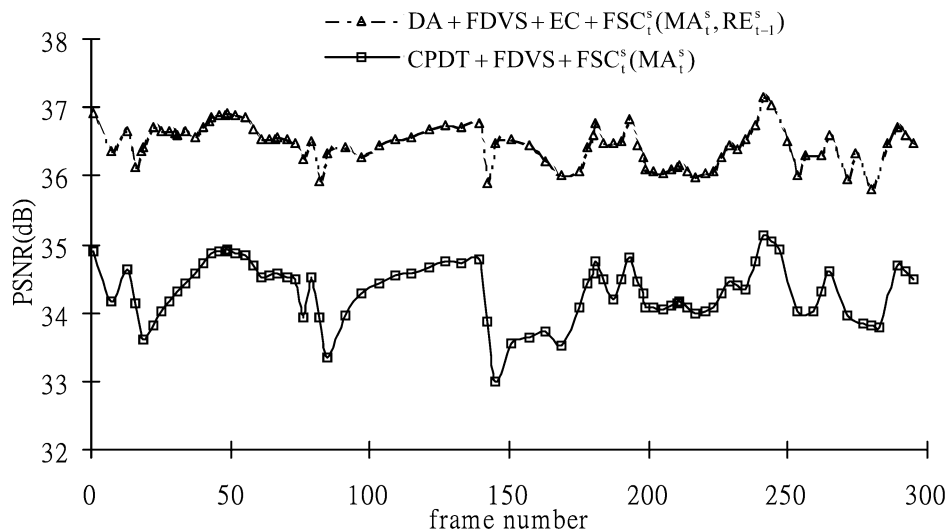


Fig. 12. PSNR of the proposed dynamic frame-skipping transcoder. The “Table Tennis” sequence was encoded by MPEG2 TM5 [27] using a coding pattern “IBBPBBP ...” with a bitrate of 3 Mb/s. The incoming frame rate was 30 frames/s, and the frame rate after transcoding was 7.5 frames/s with 0.75 Mb/s.

TABLE IX
AVERAGE PSNR AND SPEED-UP RATIO OF THE PROPOSED DYNAMIC TRANSCODER AS COMPARED WITH DA+FDVS+EC+FSC_i^s(MA_i^s). THE FRONT ENCODER WAS MPEG2 TM5 [27] WITH GOP STRUCTURE OF “IBBPBBP ...”

Sequences	Input bitrate	Output bitrate	CPDT + FDVS + FSC _i ^s (MA _i ^s)	DA + FDVS + EC + FSC _i ^s (MA _i ^s , RE _{i-1} ^s)	
			Average PSNR	Average PSNR	Speed-up ratio
Table Tennis (352x240)	1.5M	0.375M	31.26	33.95	7.05
	3M	0.75M	34.33	36.66	7.38
Football (352x240)	1.5M	0.375M	29.81	32.38	4.06
	3M	0.75M	33.73	35.98	4.17
Stefan (720x480)	7M	1.7M	30.61	33.11	4.32
	14M	3.5M	33.93	36.26	4.45

2) a cache subsystem for motion-compensated macroblocks to reduce the redundant IDCT and inverse quantization. Furthermore, we have also shown that a direct addition of the DCT coefficients on macroblocks without motion compensation and error compensation on motion-compensated macroblocks can reduce significantly the re-encoding errors due to transcoding. The overall performance of the proposed architecture produces a better picture quality than the conventional frame-skipping transcoder at the same reduced bitrates.

Furthermore, our proposed frame-skipping transcoder can be processed in the forward order when multiple frames are dropped. Thus, only one DCT-domain buffer is needed to store the updated DCT coefficients of all skipped frames. By using such a mechanism, a new frame-rate control scheme for the proposed transcoder is also suggested in this paper. Since the quality of the nonskipped frame impacts directly the motion smoothness of the transcoded sequence, it is beneficial to force the frame-rate control scheme to select frames which have good quality for reconstruction. The proposed scheme can dynamically adjust the number of skipped frames depending upon re-encoding errors as well as the accumulated magnitude of all of the motion vectors in the current frame. Experimental results show that our proposed dynamic frame-rate control scheme provides a decoded sequence with a smoother motion as well as better transcoded pictures.

REFERENCES

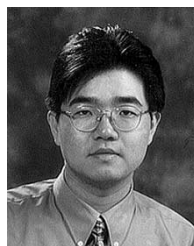
- [1] M. S. M. Lei, T. C. Chen, and M. T. Sun, “Video bridging based on H.261 standard,” *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 4, pp. 425–437, Aug. 1994.
- [2] *ITU-T Recommend. H.263, Video coding for low bitrate communication*, May 1997.
- [3] L. Chiariglione, “The development of an integrated audiovisual coding standard: MPEG,” *Proc. IEEE*, vol. 83, pp. 151–157, Feb. 1995.
- [4] *ISO/IEC 11 172-2, Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s—Part 2: Video*, 1993.
- [5] *ISO/IEC 13 818-2, Information technology—Generic coding of moving pictures and associated audio information: Video*, 1996.
- [6] H. J. Stuttgart, “Network evolution and multimedia communication,” *IEEE Trans. Multimedia*, vol. 2, pp. 42–59, Fall 1995.
- [7] C.-W. Lin, T.-J. Liou, and Y.-C. Chen, “Dynamic rate control in multi-point video transcoding,” in *Proc. IEEE Int. Symp. Circuits and Systems 2000*, vol. 2, May 28–31, 2000, pp. 17–20.
- [8] G. Keeman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, “Transcoding of MPEG-2 bitstreams,” *Signal Process.: Image Commun.*, vol. 8, pp. 481–500, Sept. 1996.
- [9] J. Youn, M.-T. Sun, and C.-W. Lin, “Motion vector refinement for high-performance transcoding,” *IEEE Trans. Multimedia*, vol. 1, pp. 30–40, Mar. 1999.
- [10] —, “Motion estimation for high performance transcoding,” *IEEE Trans. Consumer Electron.*, vol. 44, pp. 649–658, Aug. 1998.
- [11] J.-N. Hwang, T.-D. Wu, and C.-W. Lin, “Dynamic frame-skipping in video transcoding,” in *1998 IEEE 2nd Workshop on Multimedia Signal Processing*, 1998, pp. 616–621.
- [12] H. Sun, W. Kwok, and J. W. Zdepski, “Architectures for MPEG compressed bitstream scaling,” *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 6, pp. 191–199, Apr. 1996.

- [13] Y. Nakajima, H. Hori, and T. Kanoh, "Rate conversion of MPEG coded video by re-quantization process," in *Proc. IEEE Int. Conf. Image Processing '95*, vol. 3, Washington, DC, Oct. 1995, ICIP95, pp. 408–411.
- [14] P. Assuncao and M. Ghanbari, "Post-processing of MPEG2 coded video for transmission at lower bit rates," in *Proc. IEEE Int. Conf. on Acoustic, Speech, and Signal Processing '96*, vol. 4, Atlanta, GA, May 1996, pp. 1998–2001.
- [15] Y. L. Chan and W. C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 6, pp. 113–118, Feb. 1996.
- [16] —, "Edge oriented block motion estimation for video coding," *Proc. Inst. Elect. Eng.*, vol. 144, no. 3, pp. 136–144, June 1997.
- [17] —, "On block motion estimation using a novel search strategy for an improved adaptive pixel decimation," *J. Vis. Commun. Image Represent.*, vol. 9, no. 2, pp. 139–154, Jun. 1998.
- [18] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 8, pp. 369–377, Aug. 1998.
- [19] L.-M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 6, pp. 313–317, June 1996.
- [20] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 4, pp. 438–442, Aug. 1994.
- [21] K. T. Fung, Y. L. Chan, and W. C. Siu, "Low-complexity and high quality frame-skipping transcoder," in *Proc. IEEE Int. Symp. Circuits and Systems 2001*, Sydney, Australia, May 6–9, 2001, pp. 29–32.
- [22] Y.-Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding—Part II: Interframe prediction," *IEEE Trans. Broadcast.*, vol. 37, pp. 102–105, Sept. 1991.
- [23] J. Chalidabhongse and C.-C. J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 7, pp. 477–488, June 1997.
- [24] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, pp. 74–90, Nov. 1998.
- [25] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 8, pp. 849–866, Nov. 1998.
- [26] *ITU-T/SG15, Video codec test model, TMN8*, June 1997.
- [27] *ISO/IEC, Test Model 5, TMS, ISO/IEC JTC/SC29/WG11/N0400, MPEG93/457*, Apr. 1993.



Kai-Tat Fung received the B.Eng. and M.Phil. degrees from The Hong Kong Polytechnic University in 1998 and 2001, respectively; he is currently pursuing the Ph.D. degree.

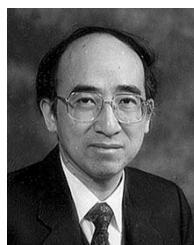
His research interests include video transcoding, video conferencing application, image and video technology, audio compression, and blind signal separation.



Yui-Lam Chan received the B.Eng. degree with First Class Honors and the Ph.D. degree from The Hong Kong Polytechnic University (HKPU) in 1993 and 1997, respectively.

He joined HKPU in 1997 and is now an Assistant Professor in the Centre for Multimedia Signal Processing and the Department of Electronic and Information Engineering. He has published over 20 research papers in various international journals and conferences. His research interests include multimedia technologies, signal processing, image and video compression, video transcoding, video conferencing, and digital TV.

Dr. Chan is the recipient of more than ten prizes, scholarships, and fellowships for his outstanding academic achievements, such as the Champion of the Varsity Competition in Electronic Design, the Sir Edward Youde Memorial Fellowship, and the Croucher Foundation Scholarships.



Wan-Chi Siu (S'77–M'77–SM'90) received the Associateship from The Hong Kong Polytechnic University (HKPU) (formerly called the Hong Kong Polytechnic), the M.Phil. degree from The Chinese University of Hong Kong, and the Ph.D. degree from Imperial College of Science, Technology and Medicine, London, U.K., in 1975, 1977, and 1984, respectively.

He was with The Chinese University of Hong Kong from 1975 to 1980. He then joined HKPU as a Lecturer in 1980 and became Chair Professor in 1992. He took up administrative duties as Associate Dean of the Engineering Faculty from 1992 to 1994 and was Head of Department of Electronic and Information Engineering from 1994 to 2000. He has been Director of the Centre for Multimedia Signal Processing and Dean of Engineering Faculty of the same university since September 1998 and September 2000, respectively. He has published over 200 research papers, and his research interests include digital signal processing, fast computational algorithms, transforms, image and video coding, and computational aspects of pattern recognition and neural networks. He is a member of the editorial board of the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* and the *EURASIP Journal on Applied Signal Processing*.

Dr. Siu was a Guest Editor of a Special Issue of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II, published in May 1998, and was also an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II between 1995–1997. He was the general chair or the technical program chair of a number of international conferences. In particular, he was a Co-Chair of the Technical Program Committee of the IEEE International Symposium on Circuits and Systems (ISCAS'97) and the General Chair of the 2001 International Symposium on Intelligent Multimedia, Video, and Speech Processing (ISIMP'2001) which were held in Hong Kong, in June 1997 and May 2001, respectively. He is now the General Chair of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'2003) which will be held in Hong Kong. Between 1991 and 1995, he was a member of the Physical Sciences and Engineering Panel of the Research Grants Council (RGC), Hong Kong Government, and in 1994 he chaired the first Engineering and Information Technology Panel to assess the research quality of 19 Cost Centers (Departments) from all universities in Hong Kong. He is a Chartered Engineer, a Fellow of the IEE and the HKIE, and has also been listed in *Marquis Who's Who in the World*, *Marquis Who's Who in Science and Engineering* and other citation biographies.