DCT-Based Video Downscaling Transcoder Using Split and Merge Technique

Kai-Tat Fung and Wan-Chi Siu

***Abstract*—For a conventional downscaling video transcoder, a video server has firstly to decompress the video, perform downscaling operations in the pixel domain, and then recompress it. This is computationally intensive. However, it is difficult to perform video downscaling in the discrete cosine transform (DCT)- domain since the prediction errors of each frame are computed from its immediate past higher resolution frames. Recently, a fast algorithm for DCT domain image downsampling has been proposed to obtain the downsampled version of DCT coefficients with low computational complexity. However, there is a mismatch between the downsampled version of DCT coefficients and the resampled motion vectors. In other words, significant quality degradation is introduced when the derivation of the original motion vectors and the resampled motion vector is large. In this paper, we propose a new architecture to obtain resampled DCT coefficients in the DCT domain by using the split and merge technique. Using our proposed video transcoder architecture, a macroblock is splitted into two regions: dominant region and the boundary region. The dominant region of the macroblock can be transcoded in the DCT domain with low computational complexity and re-encoding error can be avoided. By transcoding the boundary region adaptively, low computational complexity can also be achieved. More importantly, the re-encoding error introduced in the boundary region can be controlled more dynamically. Experimental results show that our proposed video downscaling transcoder can lead to significant computational savings as well as videos with high quality as compared with the conventional approach. The proposed video transcoder is useful for video servers that provide quality service in real-time for heterogeneous clients.**

***Index Terms*—DCT-domain transcoder, downscaling, drift elimination , transcoding, video coding.**

I. Introduction

VIDEO transcoding becomes an important role for a video server to provide quality support services to heterogeneous clients or transmission channels [1]–[13]. It is in this scenario that the video server should have the capability of performing transcoding using different transcoding approaches. It converts a previously compressed video bitstream into a lower bit-rate bitstream without modifying its original structure according to the client devices (e.g., mobile phones) in terms of its computational complexity and bandwidth constraint. Several transcoding approaches are available. These include requantization [3]–[5], frame rate reduction [6]–[9], and video downscaling [10]–[13]. In this paper, we will focus on techniques for video downscaling. One straightforward approach for implementing transcoding is to cascade a decoder and an encoder [4], commonly known as pixel-domain transcoding. That is, we have to downscale an encoded video produced by one of the current video compression standards [14]–[17] such as MPEG, H.261, or H.263, which employ motion compensated prediction to exploit the temporal redundancy to achieve low bit rates. The conventional approach needs to decompress the video and perform downscaling of the video in the pixel domain [4]. Then new motion vectors and discrete cosine transform (DCT) coefficients of this downscaled video have to be recomputed inside a transcoder. This involves high computational complexity, large memory size, and long delay on a video server. One simple approach to reduce the computational complexity is to take the average of four motion vectors of the associated four macroblocks and downscale it by two so that a resampled motion vector for the downscaled version of the video can be obtained. Motion vectors obtained in this manner are not optimal [10]. As a consequence, some information reusing approaches [10]–[13], such as the adaptive motion vector resampling [10], were suggested to provide an efficient solution to recompose new motion vectors. However, these approaches only deal with the problem of motion re-estimation during the transcoding process and DCT coefficients with lower resolution are required to recompute. Due to a mismatch between the resultant motion vector and the incoming DCT coefficients, the video transcoder has to recalculate the new DCT coefficients with lower resolution from the pixel domain; this can create undesirable complexity as well as introduce re-encoding errors. A detailed analysis of this problem will be given in Section II.

Video downscaling techniques in the pixel domain for bit-rate reduction of compressed video have been studied in recent years [10]–[13]. For instance, the video downscaling transcoder proposed in [10] made use of an adaptive motion vector resampling scheme when frame-size conversion is needed. The resampling scheme suggested to align the weighting toward the worst prediction to recompose an outgoing motion vector from the incoming motion vectors of the incoming frame which has a higher resolution. In [10], a hybrid adaptive motion vector resampling (AMVR) system was proposed to downscale the video such that the transcoded sequence can avoid full motion re-estimation. These techniques are useful for video downscaling

Manuscript received April 8, 2003; revised February 7, 2005. This work was supported by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University and the Research Grant Council (CERG-Q708) of the Hong Kong SAR Government. K.T. Fung acknowledges the research studentships provided by the University. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Trac D. Tran.

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

Digital Object Identifier 10.1109/TIP.2005.863118

transcoders in the pixel-domain. That is, the resultant architecture performs the transcoding operations in the pixel domain mainly. This is a process which involves inverse DCT, DCT, quantization, and requantization process. The process can lead to high computational complexity, as well as introduce re-encoding errors.

Recently, the DCT domain transcoding has been introduced [18]–[20]. In this case, the incoming video bitstream is partially decoded to form the DCT coefficients which are then downscaled in the DCT domain. Since the 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. DCT-domain transcoding is a very attractive approach for many video applications. However, it is difficult to perform video downscaling in the DCT-domain since the prediction errors of each frame are computed from its immediate past higher resolution frames with motion compensation. In other words, quantized DCT coefficients of the residual signal with lower resolution are no longer valid because only DCT coefficients of the residual signal with higher resolution are available. The problem of this approach, however, is that re-encoding errors are introduced when the incoming motion vectors are not the same. In other words, errors will be introduced in the nonoverlapping region of the macroblock caused by the difference between the resampled motion vector and the original motion vector. A careful analysis of the situation will be given in the following sections. This mismatch of the prediction errors with the new resampled motion vector will cause poor video quality. More importantly, this re-encoding error will be accumulated and propagated to latter frames. This phenomenon is called "drift" degradation, which often results in an unacceptable video quality.

Motivated by this, we propose in this paper a computationally efficient solution to perform video downscaling in a transcoder, mainly in the DCT-domain, to simplify computational complexity and to avoid quality degradation arising from pixel-domain transcoding. In addition, an adaptive feedback control scheme is proposed, which can adaptively control the re-encoding errors due to transcoding and avoid unnecessary operations. Since the transcoding is mainly done in the DCT domain, the computational complexity and the re-encoding error can be reduced significantly. A fast algorithm is also proposed to further speed up the transcoding process. As a result, our proposed video downscaling transcoder which has an architecture of low-complexity can provide a better transcoded sequence.

The organization of this paper is as follows. Section II of this paper presents an in-depth study of the re-encoding error in the video downscaling transcoder. The proposed DCT-based video downscaling transcoder is then described in Section III. Experimental results are presented in Section IV. Finally, some concluding remarks are provided in Section V.

II. Video Downscaling in Pixel Domain Transcoding

Fig. 1 shows the well-aligned case of the motion vectors during video downscaling. In this case, four motion vectors have the same direction and magnitude. Hence we may obtain the new motion vector by using the approaches such as align-to-average weighting (AAW), align-to-best weighting (ABW), align-to-worst weighting (AWW), or adaptive motion vector resampling (AMVR). This will provide an optimal motion vector [10], as shown in Fig. 2. However, when not all the motion vectors are well aligned as shown in Fig. 3, a good motion vector resampling or motion vector refinement [4] is required in order to reduce the re-encoding error. In this case, new prediction errors have to be recomputed due to a mismatch of the DCT coefficients and the new motion vector which was estimated by using the adaptive motion vector resampling approach [10].

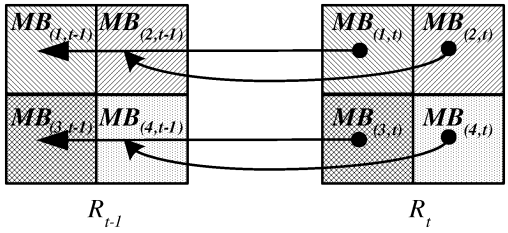

Fig. 1. All motion vectors having the same direction and magnitude.

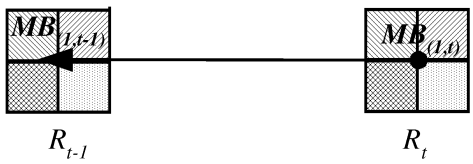

Fig. 2. Optimal motion vector after video downscaling.

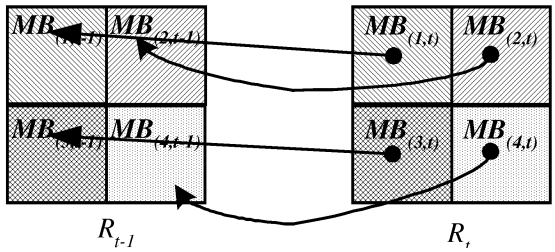

Fig. 3. Motion vectors with different directions and magnitudes.

Fig. 4 shows the architecture of the hybrid AMVR system proposed in [10]. In this hybrid system, the spatial frames are reconstructed and downscaled in the pixel domain but the motion vectors are estimated directly from the existing motion vectors in the original sequence. Initially, the variable-length decoding is performed and information of the motion vectors is extracted. Then inverse quantization and inverse DCT are performed for the incoming coefficients. After the motion compensation process, the pixel domain data are decoded and stored in the frame memory. This frame memory is used to reconstruct the next incoming frame. After the decoding process, a downscale process is applied to the decoded data in the pixel domain. In order to speed up the re-encoding process, the process of motion re-estimation is not performed. In other words, the AMVR

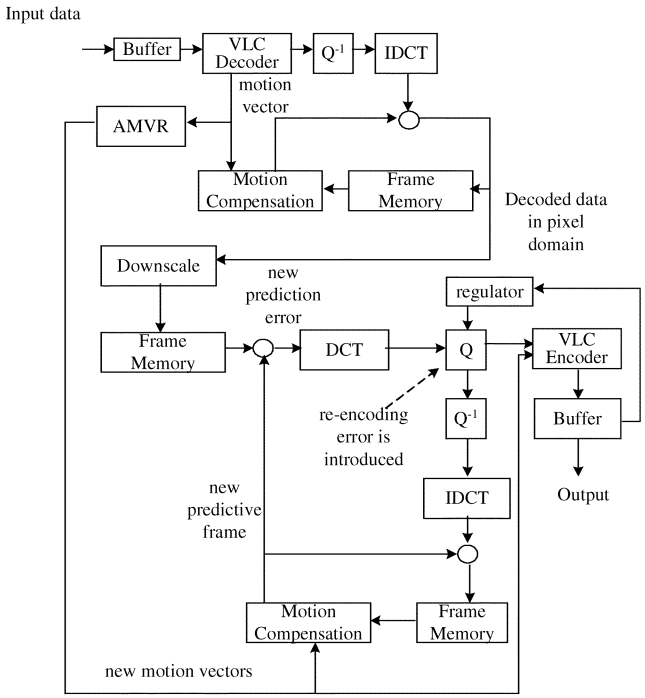

Fig. 4. Architecture of the hybrid AMVR system.

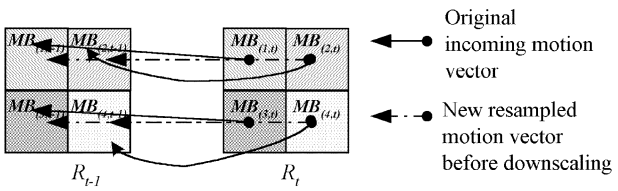

Fig. 5. New motion vector estimated before downscaling.

block is responsible for resampling of the motion vectors adaptively. The new motion vector $\bar{m}\bar{v}'$ is obtained by making use of the following equation:

$$\bar{m}\bar{v}' = \frac{1}{2}\frac{\sum_{i=1}^{4} \bar{m}\bar{v}_i A_i}{\sum_{i=1}^{4} A_i} \tag{1}$$

where $\bar{m}\bar{v}_i$ denotes the motion vector of block i in the original $N \times N$ video and A_i denotes the activity measurement of the ith residual block. The simplest way to calculate A_i is to count the number of nonzero ac coefficients. Fig. 5 shows the new motion vector before downscaling. Due to the mismatch of this new motion vector and the incoming DCT coefficients, new prediction errors need to be recomputed. Therefore, the motion compensation process is applied using the new set of motion vectors for the lower resolution frame. New prediction errors have to be calculated. Then the DCT and quantization processes are applied to the new prediction errors. During the quantization process, re-encoding errors are introduced. In order to control the accumulation of errors, inverse quantization and inverse DCT are performed. Therefore, re-encoding error can be fedback to the memory to avoid the accumulation of errors during the transcoding process. Although re-encoding errors can be controlled by using this architecture, however,

Fig. 6. Re-encoding error introduced by the video downscaling transcoder using AMVR system.

high computational complexity is required and re-encoding errors still exist in the client decoding side. The effect of re-encoding errors is shown in Fig. 6 where the "Table Tennis" sequence was transcoded at a quarter of the incoming frame size. This figure shows that re-encoding errors lead to a significant degradation of picture quality. In the next section, a DCT-based video downscaling system is proposed. Our architecture tries to transcode the video mainly in the DCT domain by reusing the incoming DCT coefficients and the incoming motion vectors. Subsequently, re-encoding errors can be reduced significantly, and a system with low computational complexity can be achieved.

III. Low Complexity and High-Quality Video Downscaling for Transcoding in the DCT Domain Using Split and Merge Technique

In this section, we present a new DCT-based video downscaling transcoding architecture. The new architecture has the following main features:

1) transcoding the overlapping regions of MC macroblocks in the DCT domain using the split and merge technique;
2) adaptively transcoding the nonoverlapping regions of MC boundary macroblocks;
3) reconstructing the new prediction errors with the architecture using the adaptive re-encoding error control;
4) fast DCT-based transcoding of MC macroblocks using significant coefficients.

The architecture of the proposed transcoder is shown in Fig. 7. The input bitstream is firstly parsed with a variable-length decoder to extract the header information, coding mode, motion vectors and quantized DCT coefficients for each macroblock. Note that each macroblock is manipulated independently. Switch SW_1 is employed to pass the reconstructed and quantized DCT coefficients to the DCT-domain downsampling operator for the transformed and quantized residual signal. The selection depends 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 I. For non-MC macroblocks or the well-align case (e.g., all motion vectors have the same magnitudes and

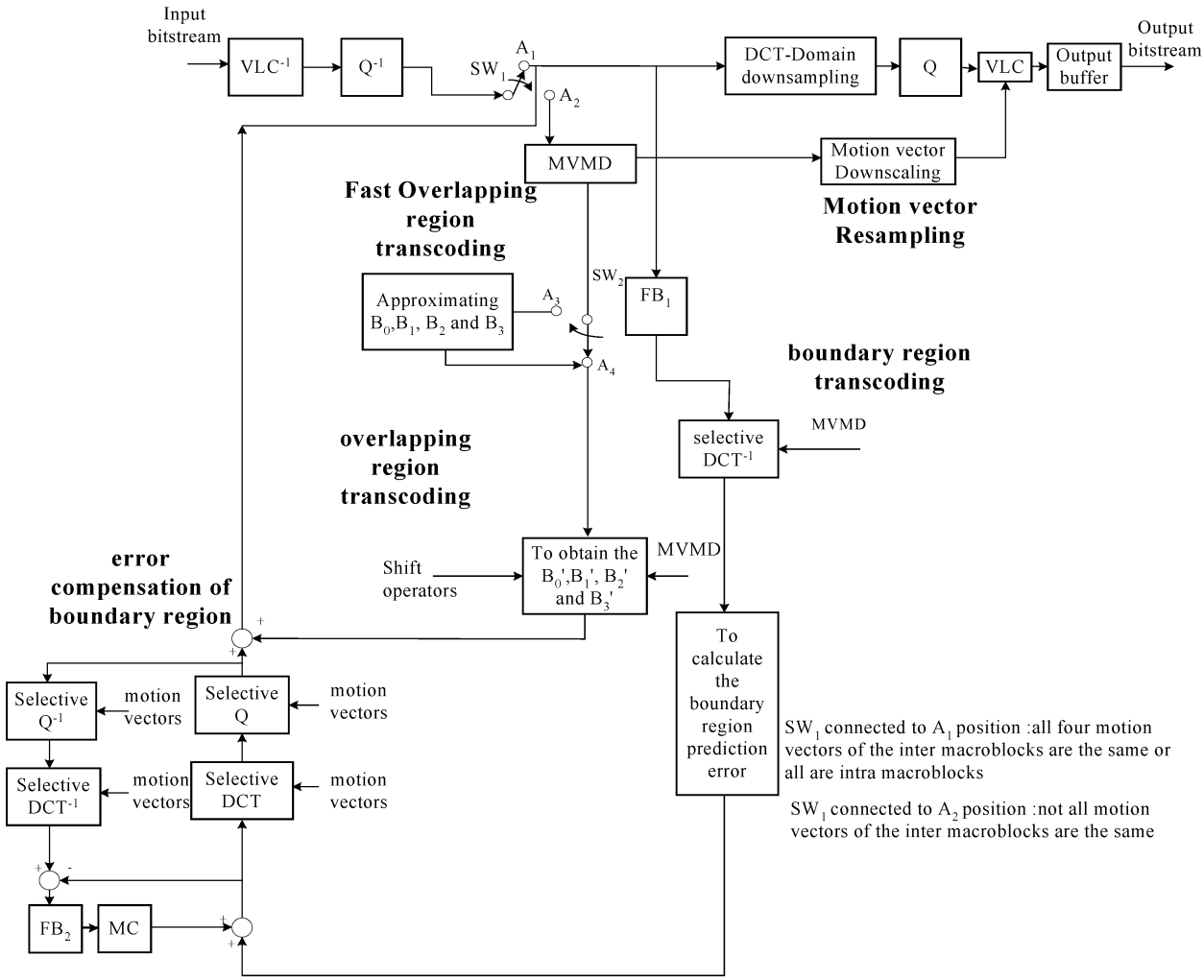

Fig. 7. Architecture proposed for DCT-based video downscaling video transcoder.

TABLE I
DIFFERENT CODING MODES OF SWITCHES SW1 OF THE PROPOSED TRANSCODER

Coding mode	SW_1 Position
Non MC/ well aligned	A_1
Not well aligned	A_2

directions) as shown in Fig. 1, the incoming prediction error in the DCT-domain is directly downsampled in the DCT domain. Hence, low computational complexity can be achieved and the quality degradation introduced using the pixel domain approach can be avoided.

When the motion vectors are not well aligned as shown in Fig. 3, direct downsampling in the DCT domain cannot be achieved since the incoming prediction errors mismatch with the reconstructed new motion vector as shown in Fig. 8. The major difficulty to transcode these MC macroblocks is that re-encoding errors will be generated due to the re-encoding process of the new DCT coefficients, which introduces quality degradation in the transcoded sequence. Also, high computational complexity is required. Motivated by this, our proposed architecture transcodes the new prediction errors mainly in the DCT domain by splitting the macroblock into overlapping region and boundary region. By calculating the new motion vector with the minimum distance (MVMD) among the four incoming motion vectors as shown in Fig. 9, the overlapping region between the incoming DCT coefficients and the target

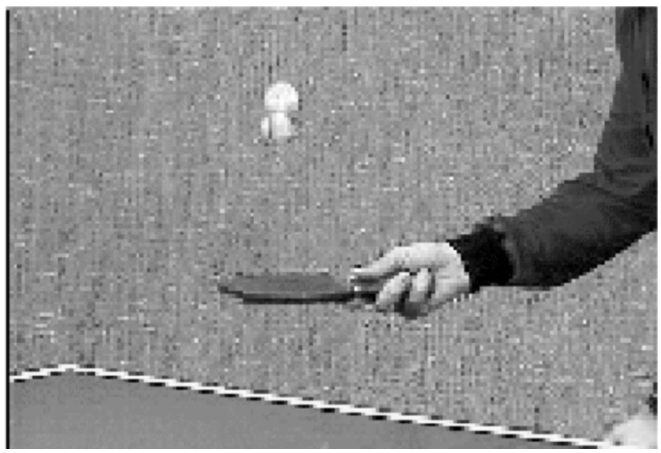

Fig. 8. Result of mismatching between the incoming DCT coefficients and the new reconstructed motion vector.

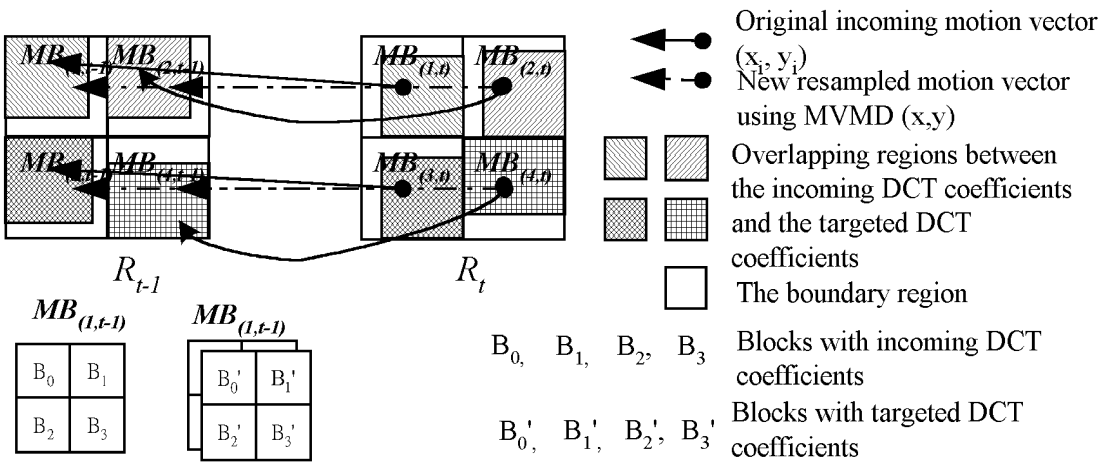

Fig. 9. Diagram showing the way to avoid video quality degradation in the overlapping region.

TABLE II
SWITCH POSITIONS FOR DIFFERENT OVERLAPPING REGION TRANSCODING APPROACHES OF THE PROPOSED TRANSCODER

Coding mode	SW_2 Position
Fast Overlapping region transcoding	A_3
use all DCT coefficients	A_4

new DCT coefficients can be reused. In other words, full inverse DCT, forward DCT, quantization, and requantization are not required in the overlapping region. Therefore, the video quality degradation in the overlapping region can be avoided and low computational complexity can be achieved. For the boundary regions, adaptive DCT, adaptive IDCT, adaptive quantization, and adaptive requantization are used to calculate the DCT coefficients. Frame buffer FB2 is proposed to feedback the re-encoding errors introduced in the boundary regions of the macroblocks. Hence, the re-encoding error introduced in the boundary regions of the macroblocks can be controlled more dynamically without introducing any redundant operations. In order to reduce the computational complexity during the MC macroblock transcoding, switch SW_2 is used to further speed up the transcoding process when fast overlapping region transcoding is employed. Table II shows different overlapping region transcoding approaches of the proposed transcoder. The advantages of the DCT-domain downscaling arrangement, together with the details of other methods, are described in the following subsections.

A. Transcoding the Overlapping Region of MC Macroblocks in the DCT Domain Using the Split and Merge Technique

For MC macroblocks, direct downscaling of the DCT coefficients cannot be employed since there is a mismatch between new resultant motion vector and the incoming DCT coefficients as shown in Fig. 5. In other words, the DCT coefficients corresponding to the new resampled motion vector are not available from the incoming bitstream. Fig. 10 shows the overlapping area of the incoming DCT coefficients. Our objective is to obtain new DCT coefficients in the $\mathrm{MB}_{(1,t-1)}$ by using parts of the four segments which come from its four neighboring macroblocks.

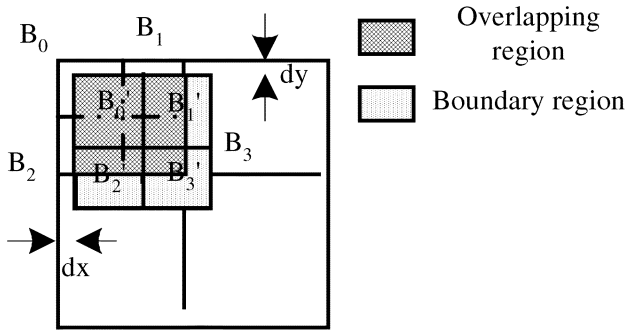

Fig. 10. Overlapping region and the boundary region of MB_{t-1}.

B_0	B_1	B_8	B_9
B_2	B_3	B_{10}	B_{11}
B_4	B_5	B_{12}	B_{13}
B_6	B_7	B_{14}	B_{15}

Fig. 11. Incoming DCT coefficients of four macroblocks.

In this paper, we split an MC macroblock in two types of regions: overlapping regions and boundary regions, as shown in Fig. 10. In the overlapping region, we propose a new minimum distance motion vector and a shift operator to compute the new DCT coefficients. This is to achieve low computational complexity and avoid re-encoding errors.

Fig. 9 shows the scenario that the current macroblock is referring to the macroblock in the previous frame. The prediction errors obtained from the incoming bitstream are $\mathrm{B}_0, \mathrm{B}_1, \ldots \mathrm{B}_{15}$ as shown in Fig. 11, where $\mathrm{B}_0, \mathrm{B}_1, \mathrm{B}_2$, and B_3 have the same motion vector whilst other blocks have three different motion vectors. Since our proposed new resampled motion vector before downscaling is different from the incoming motion vectors as shown in Fig. 9, the new prediction errors $\mathrm{B}'_0, \mathrm{B}'_1, \mathrm{B}'_2$, and B'_3 referencing to the previous frame as shown in Fig. 10 have

to be obtained in order to employ the downscaling of DCT coefficients in the DCT domain. The resampled motion vector can be obtained by minimizing the following cost functions:

$$\begin{aligned}\text{Cost\_}f(x) &= (x-x_1)^2+(x-x_2)^2+(x-x_3)^2 \\ &\quad +(x-x_4)^2 \qquad (1)\\ \text{Cost\_}f(y) &= (y-y_1)^2+(y-y_2)^2+(y-y_3)^2 \\ &\quad +(y-y_4)^2 \qquad (2)\end{aligned}$$

where x and y are resampled motion vectors in the horizontal and vertical directions respectively and $(x_1, y_1), \ldots (x_4, y_4)$ are the motion vectors from the incoming bitstream as shown in Fig. 9. Note that if all incoming motion vectors are the same, i.e., $x_1 = x_2 = x_3 = x_4$ and $y_1 = y_2 = y_3 = y_4$, the new motion vectors become (x_1, y_1). Otherwise, we need to minimize the cost function in both horizontal and vertical directions as follows:

$$\begin{aligned}\frac{d}{dx}\text{Cost\_}f(x) &= -2(x-x_1)-2(x-x_2) \\ &\quad -2(x-x_3)-2(x-x_4) \qquad (3)\\ \frac{d}{dy}\text{Cost\_}f(y) &= -2(y-y_1)-2(y-y_2) \\ &\quad -2(y-y_3)-2(y-y_4). \qquad (4)\end{aligned}$$

Let us set the derivatives to zero. We have

$$0 = -2(x-x_1)-2(x-x_2)-2(x-x_3)-2(x-x_4) \qquad (5)$$

$$0 = -2(y-y_1)-2(y-y_2)-2(y-y_3)-2(y-y_4). \qquad (6)$$

Hence, we can obtain

$$x = \frac{x_1+x_2+x_3+x_4}{4}, \quad y = \frac{y_1+y_2+y_3+y_4}{4}. \qquad (7)$$

Then, new prediction errors, $\mathrm{B}'_0, \mathrm{B}'_1, \mathrm{B}'_2$, and B'_3 can be obtained by using this resampled motion vector (x, y), as well as $\mathrm{B}_0, \mathrm{B}_1, \mathrm{B}_2, \mathrm{B}_3, \mathrm{B}_4, \mathrm{B}_5, \mathrm{B}_8, \mathrm{B}_{10}$, and B_{12} (see also Fig. 11) to avoid the re-encoding errors in the overlapping region if they can be obtained in the DCT domain directly.

Since a shifted version of the original motion vector is used, the corresponding prediction errors in the overlapping region can be obtained without performing the full re-encoding process. If the other three motion vectors have the same direction and magnitude (i.e., well-aligned), all new prediction errors ($\mathrm{B}'_0, \mathrm{B}'_1, \mathrm{B}'_2$ and B'_3) can be obtained in the DCT domain. In other words, no re-encoding error will be introduced. Otherwise, a decomposition of the overlapping region and boundary regions are needed.

The major idea to obtain the DCT coefficients in the overlapping region is to represent these coefficients as the sum of their horizontally and/or vertically displaced anchor blocks. Then the DCT values of $\mathrm{B}'_0, \mathrm{B}'_1, \mathrm{B}'_2$, and B'_3 are constructed using the pre-computed DCT values of the shift matrics.

Consider that P'_0 is the target block of interest in the pixel domain as shown in Fig. 12. $\mathrm{P}_0, \mathrm{P}_1, \mathrm{P}_2$, and P_3 are four neighboring blocks in the pixel domain from which P'_0 is derived, and the new resampled motion vector obtained by using the minimum distance derivation approach is (x, y). The minimum distance is $(\mathrm{dx} = \mathrm{x} - \mathrm{x_i}, \mathrm{dy} = \mathrm{y} - \mathrm{y_i})$ from the original one. The shaded regions in $\mathrm{P}_0, \mathrm{P}_1, \mathrm{P}_2$, and P_3 are moved by

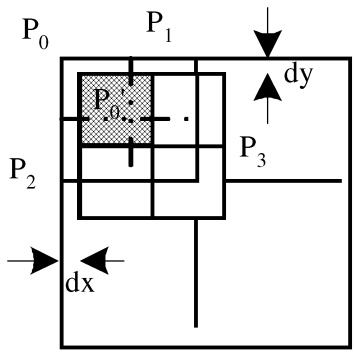

Fig. 12. Overlapping and boundary regions of MB_{t-1}.

TABLE III
SHIFTING OPERATORS

Sub-block	S_{i1}	S_{i2}
P_0	$\begin{bmatrix} 0 & I_{hd} \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ I_{wd} & 0 \end{bmatrix}$
P_1	$\begin{bmatrix} 0 & I_{hd} \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & I_{wb} \\ 0 & 0 \end{bmatrix}$
P_2	$\begin{bmatrix} 0 & 0 \\ I_{hb} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ I_{wd} & 0 \end{bmatrix}$
P_3	$\begin{bmatrix} 0 & 0 \\ I_{hb} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & I_{wb} \\ 0 & 0 \end{bmatrix}$

an amount $(\mathrm{dx}, \mathrm{dy})$. Hence P'_0 can be represented by the following equation:

$$P'_0 = \sum_{i=0}^{3} S_{i1} P_i S_{i2} \qquad (8)$$

where $\mathrm{S_{ij}}$ are matrices given in Table III.

Each I is an identity matrix of size $\mathrm{hb} = \mathrm{dy}$ or $\mathrm{hd} = 8 - \mathrm{dy}$ or $\mathrm{wb} = \mathrm{dx}$ or $\mathrm{wd} = 8 - \mathrm{dx}$. The pre-multiplication shifts the subblock of interest horizontally while the post-multiplication shifts the subblock vertically. Four possible locations of the subblock of interest: upper-left, upper-right, lower-left, and lower-right, are shown in Table III.

Let us define the 2D-DCT of an 8×8 block A as

$$\mathrm{DCT}(A) = \hat{A} = TAT^t \qquad (9)$$

where T is the 8×8 DCT matrix with entries $\mathrm{t(i, j)}$ given by

$$t(i,j) = \frac{1}{2}k(i)\cos\frac{(2j+1)i\pi}{16} \qquad (10)$$

where i represents the row index, j represents the column index, and

$$k(i) = \begin{cases} \frac{1}{\sqrt{2}}, & i = 0 \\ 1, & \text{otherwise.} \end{cases} \qquad (11)$$

Recall a property of the DCT

$$\mathrm{DCT(CD)} = \mathrm{DCT(C)DCT(D)} \qquad (12)$$

where C and D are 8×8 matrices.

From (8), we have

$$\mathrm{DCT}(P_0') = \sum_{i=0}^{3} \mathrm{DCT}(S_{i1})\mathrm{DCT}(P_i)\mathrm{DCT}(S_{i2}). \quad (13)$$

This implies that $B_0' = \sum_{i=0}^{3} \mathrm{DCT}(S_{i1})(B_i)\mathrm{DCT}(S_{i2})$ Note that the DCT of S_{i1} and S_{i2} can be pre-computed and B_i can be extracted from the incoming bitstream, so the amount of re-encoding can be reduced and the computational complexity is simplified. If macroblocks $\mathrm{MB}_{(1,\mathrm{t})}, \mathrm{MB}_{(2,\mathrm{t})}, \mathrm{MB}_{(3,\mathrm{t})}$, and $\mathrm{MB}_{(4,\mathrm{t})}$ have the same motion vector, a direct downsampling of the DCT coefficients can be applied in the DCT domain. Otherwise, a decomposition of the overlapping and boundary regions are needed, as shown in Fig. 10.

Recall the following property of the DCT:

$$\mathrm{DCT}(\mathrm{A}+\mathrm{B}) = \mathrm{DCT}(\mathrm{A}) + \mathrm{DCT}(\mathrm{B}) \quad (14)$$

where A and B represent pixels in the overlapping and boundary regions with size equal to 8×8 respectively. We can split blocks $\mathrm{B}_1', \mathrm{B}_2'$, and B_3' in two regions: overlapping region and boundary region, as shown in Fig. 10.

Using (13) and (14), we have

$$\mathrm{DCT}(\mathrm{A}+\mathrm{B}) = \sum_{\mathrm{i}=0}^{3} \mathrm{DCT}(\mathrm{S_{i1}})(\mathrm{B_i})\mathrm{DCT}(\mathrm{S_{i2}}) + \mathrm{DCT}(\mathrm{B}). \quad (15)$$

Due to the difference between the incoming motion vectors and the resampled motion vector, B_4 and B_5 are not required in order to obtain the overlapping region of B_2' (see also Fig. 11). Similarly, $(\mathrm{B}_8, \mathrm{B}_{10})$ and $(\mathrm{B}_5, \mathrm{B}_{10}$, and $\mathrm{B}_{12})$ are not required for finding the overlapping regions of B_1' and B_3' respectively. Note that the DCT coefficients of $\mathrm{B}_1', \mathrm{B}_2'$, and B_3' cannot be obtained completely since the DCT coefficients as described in (15) of the boundary region, B, have not been considered yet. The DCT coefficients of boundary region B have to be obtained separately by using 1-D inverse DCT, motion compensation, forward 1D-DCT and requantization as shown in Fig. 7. In this process, re-encoding error cannot be avoided due to requantization.

B. Adaptive Transcoding the Non-Overlapping Regions of MC Boundary Macroblocks

In order to transcode the MC boundary, the boundary region is extracted. Selective quantization and selective 1D-DCT of the quantized DCT coefficients of $\mathrm{MB}_{(1,\mathrm{t}}, \mathrm{MB}_{(2,\mathrm{t})}, \mathrm{MB}_{(3,\mathrm{t})}$, and $\mathrm{MB}_{(4,\mathrm{t})}$ have to be performed in the boundary as shown in Fig. 10. Note that each macroblock composes of four 8×8 blocks in common video coding standards [14]–[17], and the DCT and quantization operations are performed on units of 8×8 blocks. When processing $\mathrm{MB}_{(1,\mathrm{t})}, \mathrm{MB}_{(2,\mathrm{t})}, \mathrm{MB}_{(3,\mathrm{t})}$, and $\mathrm{MB}_{(4,\mathrm{t})}$, only their corresponding 8×8 blocks which have pixels overlapping with $\mathrm{MB_t}$ boundary are subject to the selective inverse 1D-DCT computation. Hence, part of the inverse 1D-DCT is performed in the boundary, while the motion vector $\mathrm{mv_{t-1}}$ is needed as an input to the adaptive 1D-DCT module to control which rows or columns the 1D-DCT has to be performed as shown in Fig. 7. In most cases, this approach is able to reduce significantly the required number of column or row DCT's as compared with that of the 2D-DCT approach.

Therefore, the new DCT coefficients of the boundary region can be obtained by performing DCT operations on part of the data and putting zero values in the overlapping region. Adaptive quantization is then used to achieve low computational complexity.

C. Reconstruction of the New Prediction Errors and Adaptive Re-Encoding Error Control Architecture

After obtaining the DCT coefficients of region $\mathrm{B}, \mathrm{B}_1', \mathrm{B}_2'$ and B_3' can be reconstructed by adding DCT(A) and DCT(B) together as shown in (15). In Fig. 10, the newly quantized DCT coefficients of an MC macroblock can then be further processed by downscaling these coefficients in the DCT domain as described in Fig. 7. For the well-aligned case, downscaling the incoming DCT coefficients can be performed directly in the DCT domain. Conversely, requantization is required for the formation of new DCT coefficients in the macroblock boundary if not all the motion vectors are the same. This will introduce additional re-encoding errors.

Note that re-encoding errors are introduced in the boundary region only as shown in Fig. 10. However, these errors will degrade the quality of the reconstructed frame. Since each P-frame is used as a reference frame for the following P-frame, quality degradation will propagate to later frames in a cumulative manner. If the accumulated sum of the re-encoding errors is large, it means that the quality of the transcoded sequence is degraded significantly. These accumulated errors become significant in the sequence containing a large amount of MC macroblocks and high motion activity.

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. 7 to compensate for the re-encoding errors introduced in the boundary region. The adaptive forward and inverse 1D-DCT and adaptive 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 and the inverse 1D-DCT is performed adaptively. The re-encoding errors introduced in the boundary region can be obtained by subtracting the original signal from the recovered signal after quantization. This re-encoding error is then stored in FB_2 and fed back to latter frames to avoid the accumulation of re-encoding errors.

Since motion vectors are highly correlated in the successive frames [21]–[28], 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. Note that the feedback loop for error compensation cannot ensure the elimination of all re-encoding errors generated by MC macroblock boundary. 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 the MC macroblocks between successive frames are highly correlated.

After the reconstruction of new prediction errors, domain downsampling will be performed in the DCT domain, and the resampled motion vector will also be downscaled (e.g., half of the original incoming one). After the downscaling process, variable length encoding is applied. Then the output data are stored inside the output buffer for transmission.

D. Fast DCT-Based Transcoding on MC Macroblocks Using Significant Coefficients

Since the energy distributions of DCT blocks obtained from an incoming bitstream mainly concentrate on the low frequency region, it is beneficial to approximate the DCT coefficients using significant DCT coefficients to speed up the transcoding process as mentioned in Section III-A. The number of significant DCT coefficients can be obtained by using the following equation which defines the energy of DCT coefficients of a block with size $\mathrm{N} \times \mathrm{N}$

$$\text{energy} = \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} B^2(l,m) \tag{16}$$

$$\text{Approx} \cdot \text{energy} = \sum_{l=0}^{j} \sum_{m=0}^{k} B^2(l,m) \tag{17}$$

where $B(l,m)$ represents the lth row and mth column of the DCT coefficients. j and k represent the numbers of rows and columns to approximate the original DCT coefficients. Initially, j and k are set to zero. If the approximated energy is less than 0.9 times of the original energy, j or k will be increased by 1 until an approximation of the significant DCT coefficients are obtained. Our experimental work shows that this approach only introduces a video quality drop of about 0.06 to 0.22 dB in the MC macroblock transcoding. However, it can further increase the speed of the DCT-based transcoding about 2.82 to 4.51 times, especially when the incoming video is encoded at a low bit rate.

IV. Experimental Results

Extensive experiments have been performed to evaluate the overall efficiency of various video downscaling transcoders. In the front encoder, the first frame was encoded as an intraframe (I-frame), and the remaining frames were encoded as interframes (P-frames). Picture-coding modes were preserved during transcoding.

These experiments aim at evaluating the performances of the proposed techniques including: 1) transcoding the overlapping region of MC macroblocks in the DCT domain using the minimum distance motion vector with unchange video resolution: 2) adaptive transcoding the nonoverlapping region MC macroblocks boundary; and 3) adaptive re-encoding error control architecture when applied to the video downscaling transcoder. The front encoder was employed to encode video sequences with different spatial resolutions and motion characteristics. "Salesman," "Miss America," and "Hall" in CIF (352×288) containing low motion activities and "Tennis," "Football," and "Flower" in (352×240) containing high motion activities

TABLE IV
Simulation Conditions

Approaches / Condition	Proposed DCT-domain transcoder		Conventional pixel-domain transcoders			
	DCT+MVMD	FDCT+MVMD	CPDT+AAW	CPDT+ABW	CPDT+AWW	CPDT+AMVR
Proposed DCT-based transcoding using motion vector with minimum distance (DCT+MVMD)	ON	ON	OFF	OFF	OFF	OFF
Proposed Fast DCT –based transcoding using motion vector with minimum distance with significant coefficients approximation (FDCT+MVMD)	OFF	ON	OFF	OFF	OFF	OFF

TABLE V
Average PSNR of the Proposed Transcoder, Where the Frame Rate of the Incoming Bitstream Was 30 frames/s. MPEG2 TMN5 [29] Was Used as the Front Encoder for Encoding "Salesman," "Miss_America," "Hall," "Tennis," "Football," and "Flower"

Sequences	Input bitrate	Average PSNR difference as compared with CPDT+AAW for MC macroblock transcoding.		
		CPDT+ABW	AMVR[10]	DCT+MVMD
Salesman (352x288)	512k	0.06	0.41	1.83
	256k	0.05	0.39	1.78
Miss_America (352x288)	512k	0.09	0.39	1.74
	256k	0.07	0.36	1.71
Hall (352x288)	512k	0.11	0.42	1.62
	256k	0.08	0.38	1.56
Tennis (352x240)	3M	0.12	0.43	1.45
	1.5M	0.08	0.38	1.40
Flower (352x240)	3M	0.18	0.47	1.31
	1.5M	0.15	0.43	1.25
Football (352x240)	3M	0.21	0.50	1.25
	1.5M	0.27	0.56	1.19

were encoded by an MPEG2 TM5 front encoder [29], but only P-frames were generated. For all testing sequences, the frame-rate of the incoming bitstream was 30 frames/s.

A large number of experimental works have been done to compare the performance of our architecture with conventional pixel-domain transcoders (CPDT) employing AAW [10], ABW [10], or AMVR [10] to resample a downscaled motion vector from the incoming motion vectors of the four macroblocks. Table IV shows the simulation conditions for different transcoders examined. Detailed comparisons of the average PSNR between $\mathrm{CPDT} + \mathrm{AAW}$, $\mathrm{CPDT} + \mathrm{ABW}$, $\mathrm{CPDT} + \mathrm{AMVR}$, and our proposed DCT-based transcoder using motion vector with the minimum distance $(\mathrm{DCT} + \mathrm{MVMD})$ are given in Table V. It shows that our proposed DCT-based transcoders outperform $\mathrm{CPDT} + \mathrm{AAW}$, $\mathrm{CPDT} + \mathrm{ABW}$, and $\mathrm{CPDT} + \mathrm{AMVR}$ in all cases. These results are more significant for sequences with low motion activity because our proposed DCT-based transcoder does not introduce any re-encoding error in the overlapping region since the transcoding is performed in DCT domain. Also, Table VI shows that our proposed transcoders have a speed-up of about 3.52–4.58 times faster than that of the conventional transcoder. This is done without performing full decoding and re-encoding process. Our transcoder transcodes all MC macroblocks mainly in DCT domain. For sequences with low motion activity, such as salesman, miss_America

TABLE VI
AVERAGE PSNR AND SPEED-UP RATIO OF OUR PROPOSED TRANSCODER AS COMPARED WITH CPDT + AAW USING MPEG2 TMN5 [29] AS A FRONT ENCODER

Sequences	Input bitrate	DCT+MVMD	
		Average PSNR difference as compared with CPDT+AAW	Speed-up ratio as compared with CPDT+AAW
Salesman (352x288)	512k	1.83	4.58
	256k	1.78	4.52
Miss_America (352x288)	512k	1.74	4.47
	256k	1.71	4.42
Hall (352x288)	512k	1.62	4.21
	256k	1.56	4.17
Tennis (352x240)	3M	1.45	3.97
	1.5M	1.40	3.95
Flower (352x240)	3M	1.31	3.63
	1.5M	1.25	3.61
Football (352x240)	3M	1.25	3.54
	1.5M	1.19	3.52

TABLE VII
AVERAGE PSNR AND SPEED-UP OF THE PROPOSED TRANSCODER USING SIGNIFICANT DCT COEFFICIENTS (FDCT + MVMD) AS COMPARED WITH THE PROPOSED TRANSCODER DCT + MVMD. MPEG2 TMN5 [29] WAS USED AS THE FRONT ENCODER FOR ENCODING "SALESMAN," "MISS_AMERICA," "HALL," "TENNIS," "FOOTBALL," AND "FLOWER"

Sequences	Input bitrate	FDCT+MVMD	
		Average PSNR difference as compared with DCT+MVMD	Speed-up ratio as compared with DCT+MVMD
Salesman (352x288)	512k	-0.09	2.82
	256k	-0.06	2.88
Miss_America (352x288)	512k	-0.13	2.97
	256k	-0.10	3.06
Hall (352x288)	512k	-0.16	3.34
	256k	-0.12	3.45
Tennis (352x240)	3M	-0.18	3.89
	1.5M	-0.12	4.03
Flower (352x240)	3M	-0.20	4.21
	1.5M	-0.14	4.36
Football (352x240)	3M	-0.22	4.37
	1.5M	-0.16	4.51

and Hall sequences, the motion vectors are small due to slow motion activity. Therefore, the overlapping region is large in the MC macroblock. Hence, significant improvement can be achieved, which is about 1.56–1.83 dB, as shown in Table V. For "Table tennis," "Football," and "Flower" sequences, the average PSNR and speed-up also have significant improvement as compared with the conventional approaches. It is due to the fact that re-encoding process is performed in the pixel domain [10]. Full decoding and re-encoding processes are required for the conventional pixel domain transcoder to transcode the MC macroblocks. Hence, the computational complexity, as well as re-encoding errors become significant for transcoding these video sequences. Only a small amount of region of the MC macroblock requires full re-encoding for using our proposed video transcoder, and pixel domain transcoding occurs only in boundary regions. In other words, the transcoding process is performed mostly in the DCT domain hence quality degradation can be avoided. On the average, about 1.19–1.45 dB PSNR improvement and a speed-up of 3.52–3.97 times have been achieved, as shown in Table VI.

Table VII compares the average PSNR and complexities of our proposed transcoders, DCT + MVMD, and our proposed transcoder using significant DCT coefficients for MC macroblock transcoding in the overlapping region, FDCT + MVMD. As shown in Table VII, FDCT + MVMD gives similar performance with the DCT + MVMD and it has a slight quality degradation of about 0.06–0.22 dB. Note that the speed-up can further be increased from 2.82 to 4.51 times. Therefore, FDCT + MVMD is more suitable for video downscaling transcoders with low bit-rate applications.

We have pointed out, in Table VII, that FDCT + MVMD has a slight PSNR degradation over DCT + MVMD. This result is expected since not all DCT coefficients are used in the MC transcoding process. However, the speed-up of transcoding MC macroblocks can be further increased significantly, especially in low bit-rate applications. Since the major contents of the DCT coefficients concentrate mainly on the low-frequency part and only a few DCT coefficients are nonzero in low bit-rate video coding, the approximation is very close to the actual value and the number of significant DCT coefficients is much less than 64. In the "Salesman," the speed-up performance is about 2.82, since most of the macroblocks are coded in the non-MC mode. For the "Table Tennis," "Football," and "Flower" sequences, the average PSNR degradation using the FDCT + MVMD is about 0.12–0.22 dB. However, the speed-up of FDCT + MVMD significantly outperforms DCT + MVMD for all these sequences, which is about 3.89–4.51 times.

V. CONCLUSION

In this paper, we have proposed a new architecture for low-complexity and high quality video downscaling transcoder to solve the problem of MC macroblcoks transcoding. Its low complexity is achieved by: 1) re-using the DCT coefficients for macroblocks coded with motion compensation to deactivate most of the complex modules of the transcoder; 2) using an adaptive MC macroblock boundary to help the transcoding; 3) using selective inverse 1D-DCT, 1D-DCT, quantization, and inverse quantization for motion-compensated macroblocks in error compensation; and 4) using a fast DCT-based video downscaling transcoding approach for MC macroblocks with significant coefficients to speed up the transcoding process. Furthermore, it is shown that re-encoding errors can be reduced significantly by: 1) transcoding the overlapping region of MC macroblocks in the DCT domain using the minimum distance motion vector without changing its video resolution; 2) adaptively transcoding the nonoverlapping regions of MC boundary macroblocks; and 3) making use of an adaptively

re-encoding error control architecture. On the whole, the proposed architecture produces a picture with the quality better than that of the conventional video downscaling transcoder at the same reduced bit rates. Furthermore, low computational complexity can be achieved since only the boundary region is required to make transcoding in the pixel-domain and perform error compensation. In other words, the transcoding process can be performed in the DCT domain for most of the regions, which avoids quality degradation. Experimental results show that our proposed DCT-based video downscaling transcoder has outstanding performance to transcode MC macroblocks of various video sequences.

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, no. 4, pp. 425–437, Aug. 1994.

[2] H. J. Stuttgen, "Network evolution and multimedia communication," *IEEE Multimedia*, vol. 2, pp. 42–59, Fall 1995.

[3] G. Keeman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG-2 bitstreams," *Signal Process.: Image Commun.*, vol. 8, pp. 481–500, Sep. 1996.

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

[5] ——, "Motion estimation for high performance transcoding," *IEEE Trans. Consumer Electron.*, vol. 44, no. 4, pp. 649–658, Aug. 1998.

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

[7] J.-N. Hwang, T.-D. Wu, and C.-W. Lin, "Dynamic frame-skipping in video transcoding," in *Proc. IEEE 2nd Workshop on Multimedia Signal Processing*, 1998, pp. 616–621.

[8] C.-W. Lin, T.-J. Liou, and Y.-C. Chen, "Dynamic rate control in multipoint video transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, May 28–31, 2000, pp. 17–20.

[9] K. T. Fung, Y. L. Chan, and W. C. Siu, "New architecture for dynamic frame-skipping transcoder," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 886–900, Aug. 2002.

[10] B. Shen, I. K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video downscaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, Sep. 1999.

[11] T.-P. Tan, H. Sun, and Y. Liang, "On the methods and applications of arbitrarily downsizing video transcoding," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, vol. 1, 2002, pp. 609–612.

[12] Z. Lei and N. D. Georganas, "H.263 video transcoding for spatial resolution downscaling," in *Proc. Int. Conf. N.D. Information Technology: Coding and Computing*, 2002, pp. 425–430.

[13] J. Chalidabhongse and C.-C. J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 477–488, Jun. 1997.

[14] *Video Coding for Low Bitrate Communication*, May 1997. ITU-T Rec. H.263.

[15] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," in *Proc. IEEE*, vol. 83, Feb. 1995, pp. 151–157.

[16] 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).

[17] Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video (1996).

[18] 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.

[19] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 1–11, Jan. 1995.

[20] N. Merhav and V. Bhaskaran, "Fast algortihms for DCT-domain image downsampling and for inverse motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, Jun. 1997.

[21] Y. L. Chan and W. C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 113–118, Feb. 1996.

[22] ——, "Edge oriented block motion estimation for video coding," in *IEE Proc.: Vision, Image, Signal Processing*, vol. 144, Jun. 1997, pp. 136–144.

[23] ——, "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.

[24] 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, no. 4, pp. 369–377, Aug. 1998.

[25] 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, no. 3, pp. 313–317, Jun. 1996.

[26] 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, no. 4, pp. 438–442, Aug. 1994.

[27] Y.-Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding—Part II: Inter-frame prediction," *IEEE Trans. Broadcast.*, vol. 37, no. 3, pp. 102–105, Sep. 1991.

[28] J. Chalidabhongse and C.-C. J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 477–488, Jun. 1997.

[29] Test Model 5, TM5, ISO/IEC JTC/SC29/WG11/N0400, MPEG93/457 (1993, Apr.).

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

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

Wan-Chi Siu received the Associateship from The Hong Kong Polytechnic University and the M.Phil. degree from The Chinese University of Hong Kong in 1975 and 1977, respectively, and the Ph.D. degree from Imperial College of Science, Technology, and Medicine, London, U.K., in October 1984.

He was with The Chinese University of Hong Kong as a Tutor and later as an Engineer between 1975 and 1980. He then joined The Hong Kong Polytechnic University as a Lecturer in 1980. He was promoted to Senior Lecturer, Principle Lecturer, and Reader in 1985, 1987, and 1990, respectively, and has been Chair Professor in the Department of Electronic and Information Engineering since 1992. He was Head of Department of Electronic and Information Engineering Department and subsequently Dean of the Engineering Faculty between 1994 and 2002. He is currently the Director of the Centre for Multimedia Signal Processing. He has published over 250 research papers, over 120 of which appeared in international journals, such as IEEE TRANSACTIONS ON SIGNAL PROCESSING and is an editor of the recent book *Multimedia Information Retrieval and Management* (Springer, 2003). His research interests include digital signal processing, fast computational algorithms, transforms, wavelets, image and video coding, and computational aspects of pattern recognition.

Dr. Siu was a Guest Editor, Associate Editor, and Member of the Editorial Board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–II, *Pattern Recognition*, the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, the *EURASIP Journal on Applied Signal Processing*, in addition to other journals. He has been a keynote speaker for many international conferences, including IEEE PCM-2002 (Taiwan, R.O.C.) and the IEEE 2003 ICNNSP (Nanjing, China). He has held the position of General Chair or Technical Program Chair of many international conferences, including IEEE Society sponsored flagship conferences such as ISCAS'1997 and ICASSP'2003. 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 of the Research Assessment Exercise (RAE) to assess the research quality of 19 Cost Centers (departments) from all universities in Hong Kong. He has received many awards, including the Distinguished Presenter Award (1997), the IEEE Third Millennium Medal (2000), the Best Teacher Award (2003), the Outstanding Award in Research (2003), the Plaque for Exceptional Leadership from IEEE SPCB (2003), and the Honorable Mention Winner Award from *Pattern Recognition* (2004).