

Compressed-Domain Techniques for Error-Resilient Video Transcoding Using RPS

Yui-Lam Chan, *Member, IEEE*, Hoi-Kin Cheung, and Wan-Chi Siu, *Senior Member, IEEE*

Abstract—In video applications where video sequences are compressed and stored in a storage device for future delivery, the encoding process is typically carried out without enough prior knowledge about the channel characteristics of a network. Error-resilient transcoding plays an important role to provide an addition of resilience to the video data, where or whenever it is needed. Recently, a reference picture selection (RPS) scheme has been adopted in an error-resilient transcoder in order to reduce error effects for the already encoded video bitstream. In this approach, the transcoder learns through a feedback channel about the damaged parts of a previously coded frame and then decides to code the next P-frame not relative to the most recent, but to an older, reference picture, which is known to be error-free in the decoder. One straightforward approach of adopting RPS in error-resilient transcoding is to decode all the P-frames from the previously nearest I-frame to the current transmitted frame which is then re-encoded with a new reference frame; this can create undesirable complexity in the transcoder as well as introduce re-encoding errors. In this paper, some novel techniques are suggested for an effective implementation of RPS in the error-resilient transcoder with the minimum requirement on its complexity. All the proposed techniques will manipulate video data in the compressed domain such that the computational loading of the transcoder is greatly reduced. By utilizing these new compressed-domain techniques, we develop a new structure to handle various types of macroblocks in the transcoder which re-uses motion vectors and prediction errors from the encoded bitstream. Experimental results demonstrate that significant improvements in terms of transcoder complexity and quality of reconstructed video can be achieved by employing our compressed-domain techniques.

Index Terms—Compressed-domain processing, error-resilient transcoding, streaming video, video coding.

I. INTRODUCTION

THE efficiency of the present video coding standards [1]–[4] facilitates the use of digital videos in a great variety of applications. Many modern video services and applications [5], [6], such as DVD, video on demand (VoD), and distance learning, use pre-encoded videos for storage and

transmission rather than real-time encoded videos. In real-time video encoding, data arrives from a live source at a fixed frame rate, which must be compressed in real time and transmitted at that speed. In coding a live video for transmission, the video server might use feedback from the client decoder to adapt its coding parameters to the time-varying conditions. Such coding parameters include the channel bandwidth and the bit-error-rate of the communication systems operating in the presence of noise. On the other hand, an off-line encoder utilizes the video stream that does not arrive from a live source; rather an entire video sequence is already available. Such an approach is useful for production systems such as encoding a large video sequence into a storage device for future delivery. In this off-line mode, the time-varying characteristics of the present communication networks pose difficulties for transmitting the already encoded video bitstream. For instance, an off-line encoder can only produce a single bitstream which does not adapt to the time-varying channel conditions at the time of transmission. In other words, if the already encoded bitstream is used, the lack of flexibility makes it difficult to change the bitrate and the resilience of the bitstream.

To solve the problems of bit rate regulation and error-resilience addressed above, several video transcoding proxies, which consist of a video transcoder or set of transcoders, have been proposed recently. Many rate management skills of video transcoders that convert encoded bitstreams into lower rates have widely been investigated in the literature [7]–[12]. In addition to the rate management skills of video transcoders, the error-resilient problem happens when the encoded bitstream is transmitting over the channel in which part of the already transmitted bitstream has been corrupted. In this situation, a further need for error-resilient handling of the transcoded sequence may arise over time-varying channels. Thus, video transcoders will play an important role not only matching the transmission rates to the user requirements, but also providing the necessary protection for transcoded video bitstreams prior to their transmission.

A review of some error-resilient video transcoding methods has been given in [12], [13]. Authors in [14] suggested a rate-distortion framework with analytical models for error-resilient transcoding. By moving the error-resilience support from the source encoder to the video proxy, the models are used to characterize how corruption propagates temporally and spatially in MPEG-encoded video subject to bit errors, and they optimize the combination of spatial error resilience, temporal error resilience, and transmission bitrate. In [15], an error-resilient transcoding scheme has been proposed for general packet radio services (GPRS) mobile access network. The transcoding

Manuscript received January 16, 2008; revised September 08, 2008. First published December 09, 2008; current version published January 09, 2009. This work was supported in part by the Centre for Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University, and in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (PolyU 5123/05E). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lina J. Karam.

The authors are with the Centre for Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: enylchan@polyu.edu.hk; hoikin80@gmail.com; enwcsi@polyu.edu.hk).

Digital Object Identifier 10.1109/TIP.2008.2007560

process in the video proxy utilizes two error-resilient tools: an adaptive intra-refresh (AIR) scheme and a reference picture selection (RPS) scheme with feedback control signaling. Based on the feedback signal, these tools can reduce the effect of error propagation when users playback video. For error-resilient transcoding using AIR, the transcoder increases the temporal resilience by inserting more intra-macroblocks. The incoming bitstream is fully decoded and some inter-macroblocks are then re-encoded as intra-macroblocks. Recently, some varieties [16]–[19] of using AIR in error-resilient transcoding, which adaptively adjust the intra-refresh rate according to the video content and the channel's packet loss rate, have been proposed to protect the most important macroblocks against packet losses over wireless networks. On the other hand, the RPS scheme adopted in the error-resilient transcoder can stop temporal error propagation by allowing the transcoder to alter the reference frames, which are successfully decoded in the decoder, for motion compensation. Most of the earlier work on RPS have been studied for use in transmitting a live video [20]–[22], but have not been examined in transmitting an already encoded bitstream. In fact, the impact of adopting RPS on the complexity of an error-resilient transcoder is tremendous. Consider the example as shown in Fig. 1. In this figure, the transcoder knows that a transmission error occurs in frame $n - 1$ through the feedback channel. Since B-frames are not used as references, for the sake of simplicity, we focus our discussion on the case that the encoded bitstream contains I- and P-frames only. Since frame $n - 1$ is required to act as the reference frame for the reconstruction of the next transmitted frame, frame n , the quantized discrete cosine transform (DCT) coefficients of residual signal of frame n are no longer valid because they refer to frame $n - 1$ which has been corrupted. In order to stop error propagation, in RPS, the last frame available without errors at the decoder should be selected as a reference frame, that is, frame $n - 2$ for frame n . The transcoder then needs to decode frame 0 to frame n and perform re-encoding of frame n with an older reference frame, frame $n - 2$, which is known to be the last frame available without errors in the decoder. Thus, this straightforward pixel-domain implementation requires much higher complexity of the transcoder for these decoding and re-encoding processes. Besides, the video quality suffers from the re-encoding process, which introduces additional degradation. Since frame n is used as a reference frame for the following transmitted P-frames, quality degradation propagates to later frames and it means that the quality of the transcoded sequence is degraded significantly when RPS is operated in the error-resilient transcoder.

In this paper, we propose a compressed-domain RPS for error-resilient transcoding. This scheme is macroblock-based. It can reduce the computational requirement of the transcoder and the quality degradation of the transcoded sequence arising from re-encoding. The organization of this paper is as follows. Section II of this paper presents a macroblock (MB) viewpoint of error-resilient transcoding using RPS. The proposed techniques for computing the new motion vectors and prediction errors in RPS are then described in Sections III and IV respectively. Experimental results are presented in Section V. Finally, some concluding remarks are given in Section VI.

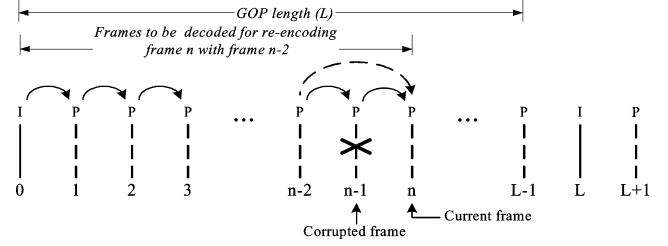


Fig. 1. Example of RPS used in error-resilient transcoding.

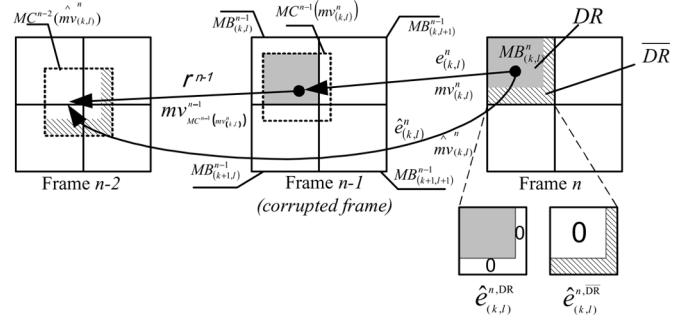


Fig. 2. MB viewpoint of RPS in error resilient transcoding. (Note: r^{n-1} is the prediction error between $MC^{n-1}(mv_{k,l}^{n-1})$ and its motion-compensated MB, which is not equal to $e_{k,l}^{n-1}$ since $MC^{n-1}(mv_{k,l}^{n-1})$ is not on a MB boundary).

II. MB VIEWPOINT OF ERROR-RESILIENT TRANSCODING USING RPS

The techniques proposed in this paper are operated at MB level. For illustration of our new MB-based techniques, the example in Fig. 1 is redrawn in MB level as depicted in Fig. 2, in which $MB_{(k,l)}^n$ represents the MB at the k th row and l th column of frame n . Assume that the transcoder receives an acknowledgment signifying that frame $n - 1$ has been damaged by transmission errors. The transcoder needs to re-encode frame n with frame $n - 2$. During decoding without transmission errors, pixels in $MB_{(k,l)}^n$ can be reconstructed by

$$MB_{(k,l)}^n = MC^{n-1}(mv_{k,l}^n) + e_{k,l}^n \quad (1)$$

where $MC^{n-1}(mv_{k,l}^n)$ stands for the motion-compensated MB of $MB_{(k,l)}^n$ which is translated by motion vector, $mv_{k,l}^n$, in the previously reconstructed frame $n - 1$, and $e_{k,l}^n$ is the prediction error between $MB_{(k,l)}^n$ and $MC^{n-1}(mv_{k,l}^n)$.

In Fig. 2, due to the transmission errors in frame $n - 1$, $MB_{(k,l)}^n$ is re-encoded with a new reference, frame $n - 2$. The new prediction error, $\hat{e}_{k,l}^n$, is given by

$$\hat{e}_{k,l}^n = MB_{(k,l)}^n - MC^{n-2}(\hat{mv}_{k,l}^n) \quad (2)$$

where $\hat{mv}_{k,l}^n$ is the new motion vector using frame $n - 2$ as a reference. The transcoder needs to compute $\hat{mv}_{k,l}^n$ and then encode $\hat{e}_{k,l}^n$ in the quantized DCT domain.

To calculate $\hat{e}_{k,l}^n$ in the pixel-domain transcoder, all the related previous MBs in P-/I-frames need to be decoded. It becomes impractical when the length of group of picture (GOP) is long. For our new MB-based techniques, two types of MBs are now defined in Fig. 3. $MB_{(k,l)}^n$ is defined as a non-motion

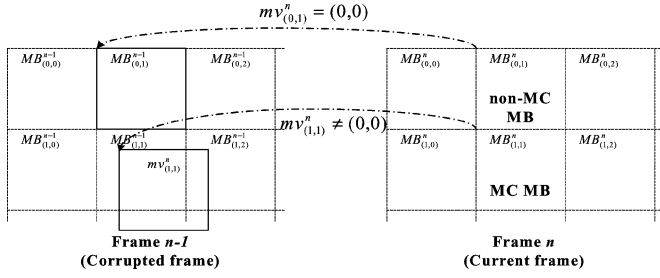


Fig. 3. Definition of non-MC and MC MBs.

compensation (non-MC) MB if the MB is coded without motion compensation. Otherwise, it is defined as a motion-compensated (MC) MB. For example, in Fig. 3, since the motion vector of $MB^{n-1}_{(0,1)}$, $mv^{n-1}_{(0,1)}$, is zero, it means that $MB^{n-1}_{(0,1)}$ is a non-MC MB. On the other hand, $MB^{n-1}_{(1,1)}$ is categorized as a MC MB. In this paper, we develop some efficient techniques, which are operated in the compressed domain, to handle various types of MBs in error-resilient transcoding with RPS in order to avoid re-encoding errors and to reduce the complexity for transcoding MBs in the affected frame. The new transcoder has three new features: 1) a quantized DCT-domain technique for non-MC MBs of the affected frame, 2) DCT-domain operators for MC MBs of the affected frame, and 3) a region tracking technique for decoding all the related MBs to the affected MC MBs. All items target at reducing the transcoder complexity, and the first one is also used for improved picture quality.

III. RE-COMPOSITION OF $\hat{mv}^n_{(k,l)}$

One simple way of getting $\hat{mv}^n_{(k,l)}$ is to perform full-scale motion estimation. This approach requires the pixel intensities of both $MB^n_{(k,l)}$ and the corresponding search area in frame $n-2$. The transcoder needs to decode all the related MBs to $MB^n_{(k,l)}$ and the corresponding search area in frame $n-2$ starting from the previously nearest I-frame. The full-scale motion estimation thus requires a high computational complexity. To reduce the computational complexity, the re-use of incoming motion vectors by adding $mv^n_{(k,l)}$ and $mv^{n-1}_{MC^{n-1}(mv^n_{(k,l)})}$ together has been widely accepted in video transcoders for bit-rate reduction [23], [24], as depicted in Fig. 2. Note that $mv^{n-1}_{MC^{n-1}(mv^n_{(k,l)})}$ is the motion vector of $MC^{n-1}(mv^n_{(k,l)})$.

Fig. 4 shows the case if $MB^n_{(k,l)}$ is found to be a non-MC MB. In this case, $mv^n_{(k,l)}$ and $mv^{n-1}_{MC^{n-1}(mv^n_{(k,l)})}$ are equal to zero and $mv^{n-1}_{(k,l)}$ respectively. Therefore, $\hat{mv}^n_{(k,l)}$ can be simplified as

$$\hat{mv}^n_{(k,l)} = mv^{n-1}_{(k,l)}. \quad (3)$$

For MC MBs, (3) does not hold true since $MC^{n-1}(mv^n_{(k,l)})$ is not on a MB boundary. In other words, $mv^{n-1}_{MC^{n-1}(mv^n_{(k,l)})}$ is not available from the encoded bitstream. In order to make an approximation of $mv^{n-1}_{MC^{n-1}(mv^n_{(k,l)})}$, it is possible to use the bilinear interpolation from the motion vectors of the four overlapping MBs with $MC^{n-1}(mv^n_{(k,l)})$ in frame $n-1$. However, the bilinear interpolation of the motion vectors can lead to the inaccuracy of the resultant motion vector when the motion vectors

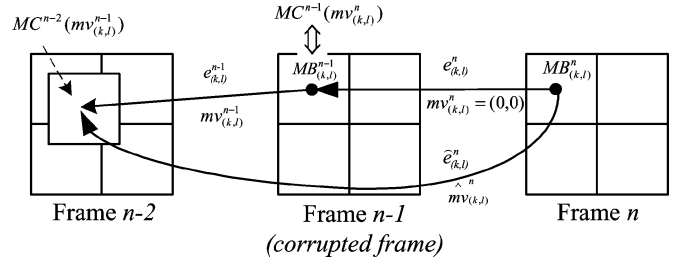


Fig. 4. RPS in a non-MC MB.

of the four overlapping MBs are too divergent and too large to be described by a single motion vector [23], [24].

Instead of using the bilinear interpolation of the available motion vectors, the forward dominant vector selection (FDVS) algorithm [23], [24], which has been well-known to provide better performance in video transcoders for bit rate reduction, is employed to speed up the error-resilient transcoding process with the adoption of RPS. As shown in Fig. 2, the FDVS algorithm is to select one dominant motion vector from the four overlapping MBs in frame $n-1$. A dominant motion vector is defined as the motion vector carried by a dominant MB. The dominant MB is the MB that has the largest overlapping segment with $MC^{n-1}(mv^n_{(k,l)})$ pointed by $mv^n_{(k,l)}$. For the example in Fig. 2, $MB^{n-1}_{(k,l)}$ is chosen as the dominant MB since it has the largest overlapping area with $MC^{n-1}(mv^n_{(k,l)})$, while its motion vector $mv^{n-1}_{(k,l)}$ is selected as the dominant motion vector. Therefore, the resultant motion vector of $MB^n_{(k,l)}$, $\hat{mv}^n_{(k,l)}$, pointing to the MB in frame $n-2$ is the sum of the dominant motion vector $mv^{n-1}_{(k,l)}$ ($mv^{n-1}_{MC^{n-1}(mv^n_{(k,l)})} = mv^{n-1}_{(k,l)}$) and $mv^n_{(k,l)}$, which can be written as

$$\hat{mv}^n_{(k,l)} = mv^{n-1}_{(k,l)} + mv^n_{(k,l)}. \quad (4)$$

IV. COMPRESSED-DOMAIN TECHNIQUES FOR COMPUTING $\hat{e}^n_{(k,l)}$

After re-composing $\hat{mv}^n_{(k,l)}$, the next step is to compute $\hat{e}^n_{(k,l)}$. By putting (1) into (2), we obtain

$$\hat{e}^n_{(k,l)} = e^n_{(k,l)} + r^{n-1} \quad (5)$$

where r^{n-1} , as shown in Fig. 2, is the prediction error between $MC^{n-1}(mv^n_{(k,l)})$ and $MC^{n-2}(\hat{mv}^n_{(k,l)})$, and can be written as

$$r^{n-1} = MC^{n-1}(mv^n_{(k,l)}) - MC^{n-2}(\hat{mv}^n_{(k,l)}). \quad (6)$$

Note that r^{n-1} is not equal to $e^n_{(k,l)}$ since $MC^{n-1}(mv^n_{(k,l)})$ is not on a MB boundary. In the MPEG-4 standard, DCT is applied to an 8×8 block and each MB is composed of four 8×8 blocks. Therefore, we can re-write (5) at block level and it is given by

$$\hat{e}^n_{(k,l),i} = e^n_{(k,l),i} + r^{n-1}_i \quad (7)$$

where $i = 1, 2, 3$, and 4, and it represents the coding order (in raster scan) of blocks within a MB. For the sake of convenience, we use the same convention for other symbols for the rest of this paper; i.e., y_i represents one of four 8×8 blocks in the MB

y_1	y_2
y_3	y_4

Fig. 5. Block order.

named y and the index i indicates the coding order of blocks within a MB, as shown in Fig. 5. Therefore, $r_1^{n-1}, r_2^{n-1}, r_3^{n-1}$, and r_4^{n-1} in (7) are the prediction errors of the four 8×8 blocks in r^{n-1} . In the following, some novel techniques for various types of MBs are proposed to compute $\hat{e}_{(k,l),i}^n$ in the compressed domain.

A. Quantized DCT-Domain Techniques for Non-MC MBs

For non-MC MBs such as $\text{MB}_{(k,l)}^n$ in Fig. 4, $\hat{mv}_{(k,l)}^n$ is equal to $mv_{(k,l)}^{n-1}$, and the spatial position of its motion-compensated MB, $\text{MC}^{n-1}(mv_{(k,l)}^n)$, in frame $n-1$ is the same as that of $\text{MB}_{(k,l)}^{n-1}$. Hence, for this specific case, $\text{MC}^{n-1}(mv_{(k,l)}^n)$ is equal to $\text{MB}_{(k,l)}^{n-1}$, and (6) can be simplified as

$$\begin{aligned} r^{n-1} &= \text{MB}_{(k,l)}^{n-1} - \text{MC}^{n-2}(mv_{(k,l)}^{n-1}) \\ &= e_{(k,l)}^{n-1}. \end{aligned} \quad (8)$$

Using (8), (7) can be written as

$$\hat{e}_{(k,l),i}^n = e_{(k,l),i}^n + e_{(k,l),i}^{n-1}. \quad (9)$$

It is noted that the quantized DCT coefficients of $e_{(k,l),i}^n$ and $e_{(k,l),i}^{n-1}$ are already available in the encoded bitstream. Let them be $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$ respectively. That is, $e_{(k,l),i}^n = \text{DCT}^{-1}(Q_L^{-1}(E_{(k,l),i}^n))$ and $e_{(k,l),i}^{n-1} = \text{DCT}^{-1}(Q_L^{-1}(E_{(k,l),i}^{n-1}))$, where $\text{DCT}^{-1}()$ is the inverse DCT operator and $Q_L^{-1}()$ is the inverse quantization operator with the quantization step size L . Therefore, (9) becomes

$$\begin{aligned} \hat{e}_{(k,l),i}^n &= \text{DCT}^{-1}\left(Q_L^{-1}\left(E_{(k,l),i}^n\right)\right) \\ &\quad + \text{DCT}^{-1}\left(Q_L^{-1}\left(E_{(k,l),i}^{n-1}\right)\right). \end{aligned} \quad (10)$$

From (10), $\hat{e}_{(k,l),i}^n$ can be computed by adding the decoded version of $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$. To stop the error propagation, the encoded form of $\hat{e}_{(k,l),i}^n$, $Q_L(\text{DCT}(\hat{e}_{(k,l),i}^n))$, is transmitted. To do so, $\hat{e}_{(k,l),i}^n$ needs to undergo transformation and quantization. Although the transcoder only needs to decode quantized DCT coefficients of two 8×8 blocks for each block in the non-MC MB with a new reference instead of decoding all its related previous MBs in P/I-frames, the additional quantization

process can lead to requantization errors, but this can be avoided if $Q_L(\text{DCT}(\hat{e}_{(k,l),i}^n))$ is computed in the quantized DCT domain, which will be discussed as shown below.

By applying DCT to (10) and taking into account the linearity of DCT, (10) can be written as

$$\text{DCT}\left(\hat{e}_{(k,l),i}^n\right) = Q_L^{-1}\left(E_{(k,l),i}^n\right) + Q_L^{-1}\left(E_{(k,l),i}^{n-1}\right). \quad (11)$$

Note that dequantization is not a linear operation because of the existence of the dead zone. However, it is reasonable to approximate $Q_L^{-1}(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}) \approx Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})$ to avoid the process of requantization. Thus, we obtain the final expression of $Q_L(\text{DCT}(\hat{e}_{(k,l),i}^n))$ as

$$Q_L\left(\text{DCT}\left(\hat{e}_{(k,l),i}^n\right)\right) \approx E_{(k,l),i}^n + E_{(k,l),i}^{n-1}. \quad (12)$$

Equation (12) implies that the newly quantized DCT coefficients $Q_L(\text{DCT}(\hat{e}_{(k,l),i}^n))$ can be computed in the quantized DCT-domain by performing direct addition (DA) of $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$. Both of them can be extracted from the already encoded bitstream by performing entropy decoding only. The error-resilient transcoder combines $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$ together which is then entropy encoded. Since no complete decoding process is necessary during the generation of $Q_L(\text{DCT}(\hat{e}_{(k,l),i}^n))$, the computational complexity required for the transcoder is very low.

B. Modified Quantizer-Dequantizer Pair for Non-MC MBs

From (12), when the effect of making a linear approximation of inverse quantization is negligible, performing DA of the quantized DCT coefficients would not give any significant loss in reconstructing a video frame with a new reference. However, since in general there is no guarantee that the effect is negligible all the time, there are non-zero probabilities that the approximation may cause some loss in the reconstructed frame. From the previous derivations, it follows that $Q_L(\text{DCT}(\hat{e}_{(k,l),i}^n))$ should be applied to the decoder. The dequantization process specified in the MPEG-4 standard is performed as shown in (13), at the bottom of the page. The term $(\text{sign}(x))/2$ is required to deliver the centroid representation of the dequantizer by introducing an addition or subtraction of half of the quantization step size, which depends on the polarity of the quantized DCT coefficient.

If the linear approximation of inverse quantization is used, we obtain the output of the dequantizer by substituting $E_{(k,l),i}^n + E_{(k,l),i}^{n-1}$ into (13), and it yields

$$\begin{aligned} Q_L^{-1}\left(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}\right) &= \left(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}\right) \\ &\quad \times L + \text{sign}\left(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}\right) \times \frac{L}{2}. \end{aligned} \quad (14)$$

$$Q_L^{-1}(x) = \begin{cases} 0, & \text{if } x = 0 \\ \left(x + \frac{\text{sign}(x)}{2}\right) \times L, & \text{otherwise} \end{cases} \quad \text{where } \text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (13)$$

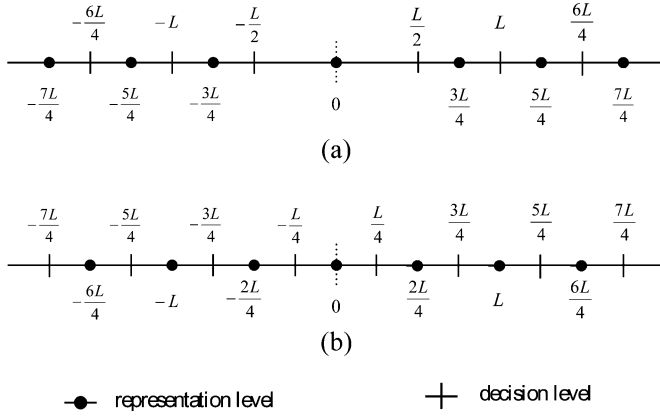


Fig. 6. Decision levels (vertical line) and representation levels (dot) of the quantizer (a) with a dead zone and (b) without a dead zone. Both of them have a quantization step size of $L/2$.

On the other hand, the output of the dequantizer without linear approximation is $Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})$ which can be actually written as

$$\begin{aligned} & Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1}) \\ &= (E_{(k,l),i}^n + E_{(k,l),i}^{n-1}) \\ &\quad \times L + \left[\text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1}) \right] \times \frac{L}{2}. \end{aligned} \quad (15)$$

Comparing (14) and (15), they are not equal since $\text{sign}(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}) \neq \text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1})$. The possible outputs of $\text{sign}(E_{(k,l),i}^n + E_{(k,l),i}^{n-1})$ are $\{-1, 0, +1\}$ while the possible outcomes of $\text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1})$ are $\{-2, -1, 0, +1, +2\}$. For example, when $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$ are non-zero and with the same sign, $\text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1})$ is either equal to 2 or -2 whereas $\text{sign}(E_{(k,l),i}^n + E_{(k,l),i}^{n-1})$ is either equal to 1 or -1. This implies that the reconstructed quality of using the DA technique in RPS with the linear approximation of inverse quantization deviates from the output without linear approximation and results in a quality drop of the transcoded sequence. The effect of approximation errors will be given in Section V.

In order to avoid the approximation errors, we need to design a better way to quantize $Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})$ and then transmit it to the decoder. If the optimum performance is desired, the two steps on quantization and dequantization have to be designed jointly. From (15), $Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})$ is composed of two terms, $(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}) \times L$ and $[\text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1})] \times (L/2)$. The former one is divisible by L . However, possible outputs of the latter one are $\{-L, -(L/2), 0, +(L/2), +L\}$, some of which are not divisible by L . In certain cases, such as when $\text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1})$ is equal to -1 or 1, the requantization with a quantization step size of L can lead to additional errors. One possible way to avoid the requantization errors is to use a step size of $L/2$ instead, since both of the terms in (15) are divisible by $L/2$. Besides, in MPEG-4, the dead zone is adopted in the quantization process. The dead zone commonly

refers to the central region of the quantizer, whereby the coefficients are quantized to zero and it is intended primarily to affect more non-significant coefficients to become zero resulting in an increase of the coding efficiency. Fig. 6(a) shows the decision levels and their corresponding representation levels of the quantizer with a quantization step size of $L/2$ and a dead zone. Since the dead zone exists, the representation levels of this quantizer become $\{\dots, -(5L/4), -(3L/4), 0, +(3L/4), +(5L/4), \dots\}$, which are not the multiple of $L/2$. That is, the representation does not match the desired representation levels of $Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})$. In Fig. 6(b), it shows the decision levels and their corresponding representation levels of the quantizer without a dead zone, again, the quantization step size is equal to $L/2$. In this case, it exactly matches the possible outcome of $Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})$. Taking all of the above considerations, the new quantized DCT coefficients with a new reference is obtained by

$$\begin{aligned} & \tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^n)) \\ &= \tilde{Q}_{L/2}(Q_L^{-1}(E_{(k,l),i}^n) + Q_L^{-1}(E_{(k,l),i}^{n-1})) \\ &= 2(E_{(k,l),i}^n + E_{(k,l),i}^{n-1}) \\ &\quad + \text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1}) \end{aligned} \quad (16)$$

where $\tilde{Q}_{L/2}()$ is the quantization operator without a dead zone and its quantization step size is equal to $L/2$. Similar to the DA technique of the quantized DCT coefficients, (16) signifies that the newly quantized DCT coefficients $\tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^n))$ can be generated in the quantized DCT domain since both $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$ can be extracted from the encoded bitstream by performing entropy decoding. The transcoder doubles the sum of $E_{(k,l),i}^n$ and $E_{(k,l),i}^{n-1}$ and then adds it to $\text{sign}(E_{(k,l),i}^n) + \text{sign}(E_{(k,l),i}^{n-1})$ which is then entropy encoded. Again, requantization is not necessary to be carried out in the formation of $\tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^n))$.

Since $\tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^n))$ is generated from the quantizer without a dead zone, the term $(\text{sign}(x)/2) \times L$ is not necessary in the dequantization process and it can be modified as shown in (17) during the decoding process of non-MC MBs

$$\tilde{Q}_{L/2}^{-1}(x) = \begin{cases} 0, & \text{if } x = 0 \\ \frac{x \times L}{2}, & \text{otherwise.} \end{cases} \quad (17)$$

Using the modified quantizer-dequantizer pair without a dead zone, the decoder can perfectly reconstruct the non-MC MBs encoded with a new reference frame. For the proposed technique, only a small change is needed for the decoder to equip with the dequantization process as shown in (17). It is exactly the MPEG dequantization equation for decoding intra-blocks with half of the quantization factor. During the transmission of $\tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^n))$, MBs that use the modified dequantizer without a dead zone have to be notified by the decoder. We note that this notification has to be transmitted as side information in the MPEG-4 video bitstream by the user data defined in [3].

The derivation in (16) assumes that the round-trip delay (RTD) corresponds to the duration of encoding time for one frame. For RTD longer than one frame, the concept can be

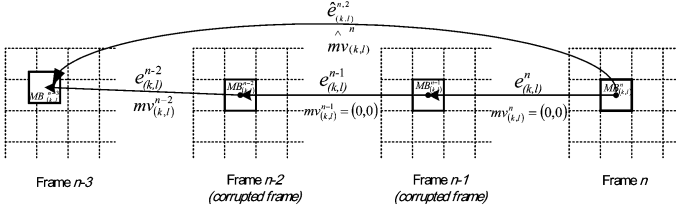


Fig. 7. Proposed techniques for non-MC MBs when RTD corresponds to the duration of encoding time for 2 frames.

further extended if the motion vectors of MBs that have the same spatial location in the skipped frames are all equal to zero. From (16), it can be easily shown that

$$\tilde{Q}_{L/2} \left(\text{DCT} \left(\hat{e}_{(k,l),i}^{n, \text{RTD_dist}} \right) \right) = \sum_{\text{RTD}=0}^{\text{RTD_dist}} \left[2E_{(k,l),i}^{n-\text{RTD}} + \text{sign} \left(E_{(k,l),i}^{n-\text{RTD}} \right) \right] \quad (18)$$

where $\tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^{n, \text{RTD_dist}}))$ is the newly quantized DCT coefficients of $\text{MB}_{(k,l)}^n$ using frame $n - \text{RTD_dist}$ as a new reference frame. Note that the index RTD_dist indicates the number of frames corresponding to the duration of encoding time at one particular RTD. Fig. 7 shows an example when RTD_dist is equal to 2. Assume that frame n is the current frame and the transcoder has been informed by the client that frame $n - 2$ is corrupted. Since the RTD is equivalent to the duration of encoding time of two frames, the transcoder then needs to generate the quantized DCT coefficients of frame n with a new reference, frame $n - 3$. Fig. 7 shows also the situation in MB level in which the motion vectors of $\text{MB}_{(k,l)}^n$ and $\text{MB}_{(k,l)}^{n-1}$ are both equal to zero. The newly quantized DCT coefficients, $\tilde{Q}_{L/2}(\text{DCT}(\hat{e}_{(k,l),i}^{n,2}))$, can be computed according to (18) which is again operated in the quantized DCT domain, thus eliminating requantization errors and the unnecessary operations for the re-encoding process.

C. Transform-Domain Operations for MC MBs

In MC MBs, the above two techniques cannot be employed since $\text{MC}^{n-1}(mv_{(k,l)}^n)$ is not equal to $\text{MB}_{(k,l)}^{n-1}$, as shown in Fig. 2. In other words, r^{n-1} is no longer equal to $e_{(k,l)}^{n-1}$ in (8). To compute $\hat{e}_{(k,l),i}^n$ of MC MBs, one obvious way is to examine the motion vectors in the already encoded bitstream and all the related MBs from the previous nearest I-frame to frame $n - 1$ should be decoded and perform re-encoding for those MC MBs with frame $n - 2$ as the reference. In this paper, the new error-resilient transcoder can minimize the use of this pixel-domain re-encoding by applying some DCT-domain operators [25], [26].

In Fig. 2, $\text{MC}^{n-1}(mv_{(k,l)}^n)$ is generally formed using parts of four segments which come from its four neighboring MBs- $\text{MB}_{(k,l)}^{n-1}$, $\text{MB}_{(k,l+1)}^{n-1}$, $\text{MB}_{(k+1,l)}^{n-1}$, and $\text{MB}_{(k+1,l+1)}^{n-1}$. Our focus here is to investigate whether it is possible to re-use the DCT coefficients of these four neighboring MBs to come up with $\text{DCT}(\hat{e}_{(k,l),i}^n)$. To avoid pixel-domain re-encoding, we

divide each MC MB into two regions: dominant region (DR) and non-dominant region ($\overline{\text{DR}}$). In Fig. 2, the areas filled with shaded color and diagonal lines are DR and $\overline{\text{DR}}$, respectively. The DR includes the pixels in $\text{MB}_{(k,l)}^n$ that point to the dominant MB in the previously corrupted frame while the remaining pixels constitute the $\overline{\text{DR}}$. Therefore, $\hat{e}_{(k,l)}^n$ can be represented by

$$\hat{e}_{(k,l)}^n = \hat{e}_{(k,l)}^{n, \text{DR}} + \hat{e}_{(k,l)}^{n, \overline{\text{DR}}} \quad (19)$$

where $\hat{e}_{(k,l)}^{n, \text{DR}}$ and $\hat{e}_{(k,l)}^{n, \overline{\text{DR}}}$ are the prediction errors in the DR and $\overline{\text{DR}}$ respectively, and these prediction errors are expanded and padded with zero such that they have the same size as $\hat{e}_{(k,l)}^n$ (8×8 in this case), as illustrated in Fig. 2. By taking into account of the linearity of DCT and applying DCT to (19), it can be written as

$$\text{DCT} \left(\hat{e}_{(k,l)}^n \right) = \text{DCT} \left(\hat{e}_{(k,l)}^{n, \text{DR}} \right) + \text{DCT} \left(\hat{e}_{(k,l)}^{n, \overline{\text{DR}}} \right). \quad (20)$$

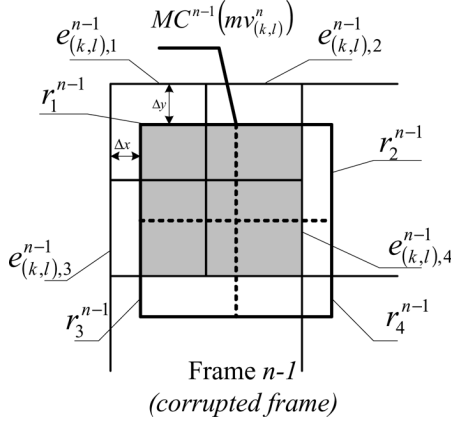
For $\text{DCT}(\hat{e}_{(k,l)}^{n, \text{DR}})$, we propose to use DCT-domain operators to achieve low computational complexity. On the other hand, pixel-domain re-encoding is still needed in $\text{DCT}(\hat{e}_{(k,l)}^{n, \overline{\text{DR}}})$. With the help of FDVS, the DR occupies the dominant area, and we can then compute $\text{DCT}(\hat{e}_{(k,l)}^{n, \text{DR}})$ as much as possible in the DCT form in order to keep the benefits of the DCT-domain manipulation.

To compute $\text{DCT}(\hat{e}_{(k,l)}^{n, \text{DR}})$ in the DCT domain, the new DCT-domain operators should be designed in block level to compute $\text{DCT}(\hat{e}_{(k,l),i}^n)$, where $i = 1, 2, 3$, and 4. From (7), by considering only the DR and applying DCT on both sides, we obtain

$$\text{DCT} \left(\hat{e}_{(k,l),i}^{n, \text{DR}} \right) = \text{DCT} \left(e_{(k,l),i}^{n, \text{DR}} \right) + \text{DCT} \left(r_i^{n-1, \text{DR}} \right) \quad (21)$$

where $e_{(k,l),i}^{n, \text{DR}}$ is the prediction error containing the DR of $\text{MB}_{(k,l)}^n$ and $r_i^{n-1, \text{DR}}$ is the prediction error overlapping with the dominant MB in frame $n - 1$. Again, $e_{(k,l),i}^{n, \text{DR}}$ and $r_i^{n-1, \text{DR}}$ are expanded to 8×8 block sizes of zero padded. Our objectives are to acquire $\text{DCT}(r_i^{n-1, \text{DR}})$ and $\text{DCT}(e_{(k,l),i}^{n, \text{DR}})$ in the DCT domain from frame $n - 1$ and frame n respectively. Although both of them cannot be retrieved directly from the encoded bitstream, they can be obtained from the existing prediction errors stored in the bitstream through the following new DCT-domain operators.

1) *Shifting Operator for $\text{DCT}(r_i^{n-1, \text{DR}})$* : Fig. 8 illustrates an example showing the detailed block structure of $\text{MC}^{n-1}(mv_{(k,l)}^n)$ in frame $n - 1$, where $mv_{(k,l)}^n$ is $(\Delta x, \Delta y)$. r_1^{n-1} , r_2^{n-1} , r_3^{n-1} , and r_4^{n-1} are the prediction errors of the corresponding four 8×8 blocks in $\text{MC}^{n-1}(mv_{(k,l)}^n)$ and their regions overlapping with the dominant MB in frame $n - 1$ are denoted by $r_1^{n-1, \text{DR}}$, $r_2^{n-1, \text{DR}}$, $r_3^{n-1, \text{DR}}$, and $r_4^{n-1, \text{DR}}$, (the shaded region in Fig. 8). Here, $e_{(k,l),1}^{n-1}$, $e_{(k,l),2}^{n-1}$, $e_{(k,l),3}^{n-1}$, and $e_{(k,l),4}^{n-1}$ are the prediction errors in $\text{MB}_{(k,l)}^{n-1}$. Their DCT coefficients ($\text{DCT}(e_{(k,l),1}^{n-1})$, $\text{DCT}(e_{(k,l),2}^{n-1})$, $\text{DCT}(e_{(k,l),3}^{n-1})$, and

Fig. 8. Block level in frame $n - 1$.

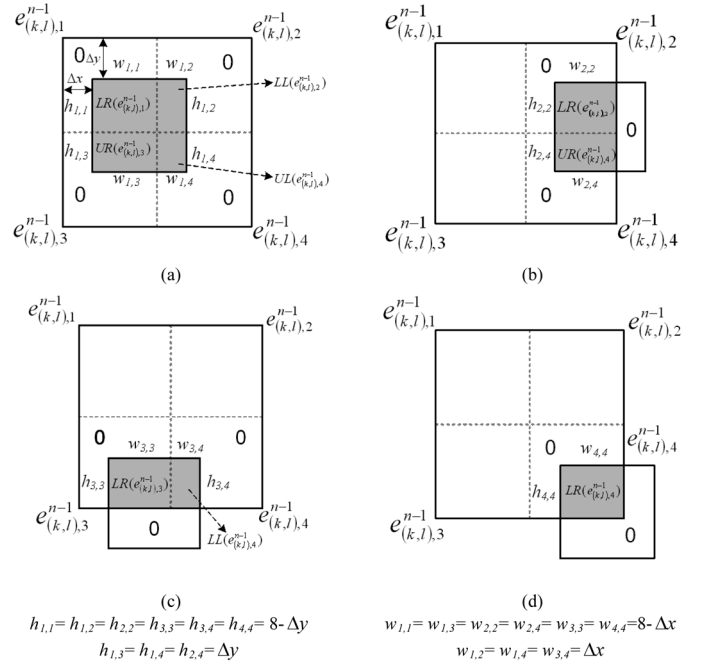
$\text{DCT}(e_{(k,l),4}^{n-1})$ are available in the encoded bitstream after performing inverse quantization. These are also the four overlapping blocks with $r_1^{n-1, \text{DR}}$, $r_2^{n-1, \text{DR}}$, $r_3^{n-1, \text{DR}}$, and $r_4^{n-1, \text{DR}}$ from which they are contributed to derive $\text{DCT}(r_i^{n-1, \text{DR}})$. From Fig. 8, it is observed that

$$r_1^{n-1, \text{DR}} = r_1^{n-1}. \quad (22)$$

In this case, $r_1^{n-1, \text{DR}}$ contains contributions from all $e_{(k,l),1}^{n-1}$, $e_{(k,l),2}^{n-1}$, $e_{(k,l),3}^{n-1}$, and $e_{(k,l),4}^{n-1}$. Specifically, it is composed of the lower-right corner of $e_{(k,l),1}^{n-1}$ ($\text{LR}(e_{(k,l),1}^{n-1})$), the lower-left corner of $e_{(k,l),2}^{n-1}$ ($\text{LL}(e_{(k,l),2}^{n-1})$), the upper-right corner of $e_{(k,l),3}^{n-1}$ ($\text{UR}(e_{(k,l),3}^{n-1})$), and the upper-left corner of $e_{(k,l),4}^{n-1}$ ($\text{UL}(e_{(k,l),4}^{n-1})$), as illustrated in Fig. 9(a). In this figure, we also supplement these contributions with zeros to form image blocks with 8×8 pixels. Then, $r_1^{n-1, \text{DR}}$ can be computed using the shifted DCT sub-blocks, as follows:

$$\begin{aligned} r_1^{n-1, \text{DR}} &= \text{LR}(e_{(k,l),1}^{n-1}) + \text{LL}(e_{(k,l),2}^{n-1}) \\ &\quad + \text{UR}(e_{(k,l),3}^{n-1}) + \text{UL}(e_{(k,l),4}^{n-1}) \\ &= \sum_{j=1}^4 s_{1,j}^{\text{ver}} e_{(k,l),j}^{n-1} s_{1,j}^{\text{hor}} \end{aligned} \quad (23)$$

where $s_{1,j}^{\text{ver}}$ and $s_{1,j}^{\text{hor}}$ are the pre-multiplication and post-multiplication matrices for computing $r_1^{n-1, \text{DR}}$, and the sub-block of interest can be shifted vertically and horizontally respectively. $s_{1,j}^{\text{ver}}$ and $s_{1,j}^{\text{hor}}$ are matrices like $\begin{pmatrix} 0 & 0 \\ I_m & 0 \end{pmatrix}$ or $\begin{pmatrix} 0 & I_m \\ 0 & 0 \end{pmatrix}$ and I_m is identity matrix of size m . The sub-block of interest generally can be classified into 4 possible locations. They are upper left, upper right, lower left, and lower right. In the equation of $r_1^{n-1, \text{DR}}$, the selection of matrices is summarized in the Table I, where the subscripts $h_{i,j}$ and $w_{i,j}$ are the number of rows and columns extracted and they are defined in Fig. 9. To obtain $\text{DCT}(r_1^{n-1, \text{DR}})$ in the DCT domain directly, we apply DCT to (23) and use the distributive property of matrix multiplication

Fig. 9. Contributions from the four overlapping blocks of (a) $r_1^{n-1, \text{DR}}$, (b) $r_2^{n-1, \text{DR}}$, (c) $r_3^{n-1, \text{DR}}$, and (d) $r_4^{n-1, \text{DR}}$.

(i.e., $\text{DCT}(AB) = \text{DCT}(A)\text{DCT}(B)$). Thus, we have

$$\begin{aligned} \text{DCT}(r_1^{n-1, \text{DR}}) &= \text{DCT} \left(\sum_{j=1}^4 s_{1,j}^{\text{ver}} e_{(k,l),j}^{n-1} s_{1,j}^{\text{hor}} \right) \\ &= \sum_{j=1}^4 S_{1,j}^{\text{ver}} E_{(k,l),j}^{n-1} S_{1,j}^{\text{hor}} \end{aligned} \quad (24)$$

where $S_{1,j}^{\text{ver}} = \text{DCT}(s_{1,j}^{\text{ver}})$ and $S_{1,j}^{\text{hor}} = \text{DCT}(s_{1,j}^{\text{hor}})$ are the shifting matrices for $\text{DCT}(r_1^{n-1, \text{DR}})$ in the DCT domain. $\text{DCT}(r_1^{n-1, \text{DR}})$ can then be extracted easily from the bitstream after the inverse VLC and de-quantization processes, which can reduce the required computation for transcoding MC MBs with a new reference.

In Fig. 9(b)–(d), $r_2^{n-1, \text{DR}}$ contains contributions from $\text{LR}(e_{(k,l),2}^{n-1})$ and $\text{UR}(e_{(k,l),4}^{n-1})$, $r_3^{n-1, \text{DR}}$ is derived from contributions of $\text{LR}(e_{(k,l),3}^{n-1})$ and $\text{LL}(e_{(k,l),4}^{n-1})$, and $r_4^{n-1, \text{DR}}$ is contributed from $\text{LR}(e_{(k,l),3}^{n-1})$ only. $r_2^{n-1, \text{DR}}$, $r_3^{n-1, \text{DR}}$, and $r_4^{n-1, \text{DR}}$ can then be computed in the same way using the suitable shifting multiplication matrices $s_{i,j}^{\text{ver}}$ and $s_{i,j}^{\text{hor}}$, as defined in Table I. Their DCT representations are also shown in Table I.

2) *Cropping Operator for $\text{DCT}(e_{(k,l),i}^{\text{DR}})$* : The DCT-domain coefficients for the second term in (21), i.e., $\text{DCT}(r_i^{n-1, \text{DR}})$, have been proven to be computed using the shifting operator. Now, a similar idea can be applied to compute $\text{DCT}(e_{(k,l),i}^{\text{DR}})$ in the DCT domain. Fig. 10 shows the detailed block structure of $\text{MB}_{(k,l)}^n$ in frame n . $\text{MB}_{(k,l)}^n$ is composed of the prediction errors, $e_{(k,l),1}^n$, $e_{(k,l),2}^n$, $e_{(k,l),3}^n$, and $e_{(k,l),4}^n$. In the encoded bitstream, their DCT coefficients ($\text{DCT}(e_{(k,l),1}^n)$, $\text{DCT}(e_{(k,l),2}^n)$, $\text{DCT}(e_{(k,l),3}^n)$, and

TABLE I
SHIFTING MATRICES FOR $r_i^{n-1,DR}$

$r_1^{n-1,DR} : DCT(r_1^{n-1,DR}) = S_{1,1}^{ver} E_{(k,l),1}^{n-1} S_{1,1}^{hor} + S_{1,2}^{ver} E_{(k,l),2}^{n-1} S_{1,2}^{hor} + S_{1,3}^{ver} E_{(k,l),3}^{n-1} S_{1,3}^{hor} + S_{1,4}^{ver} E_{(k,l),4}^{n-1} S_{1,4}^{hor}$		
$LR(e_{(k,l),1}^{n-1})$	$S_{1,1}^{ver} = \begin{pmatrix} 0 & I_{h_1} \\ 0 & 0 \end{pmatrix}$	$S_{1,1}^{hor} = \begin{pmatrix} 0 & 0 \\ I_{w_1} & 0 \end{pmatrix}$
$LL(e_{(k,l),2}^{n-1})$	$S_{1,2}^{ver} = \begin{pmatrix} 0 & I_{h_2} \\ 0 & 0 \end{pmatrix}$	$S_{1,2}^{hor} = \begin{pmatrix} 0 & I_{w_2} \\ 0 & 0 \end{pmatrix}$
$UR(e_{(k,l),3}^{n-1})$	$S_{1,3}^{ver} = \begin{pmatrix} 0 & 0 \\ I_{h_3} & 0 \end{pmatrix}$	$S_{1,3}^{hor} = \begin{pmatrix} 0 & 0 \\ I_{w_3} & 0 \end{pmatrix}$
$UL(e_{(k,l),4}^{n-1})$	$S_{1,4}^{ver} = \begin{pmatrix} 0 & 0 \\ I_{h_4} & 0 \end{pmatrix}$	$S_{1,4}^{hor} = \begin{pmatrix} 0 & I_{w_4} \\ 0 & 0 \end{pmatrix}$
$r_2^{n-1,DR} : DCT(r_2^{n-1,DR}) = S_{2,2}^{ver} E_{(k,l),2}^{n-1} S_{2,2}^{hor} + S_{2,4}^{ver} E_{(k,l),4}^{n-1} S_{2,4}^{hor}$		
$LR(e_{(k,l),2}^{n-1})$	$S_{2,2}^{ver} = \begin{pmatrix} 0 & I_{h_2} \\ 0 & 0 \end{pmatrix}$	$S_{2,2}^{hor} = \begin{pmatrix} 0 & 0 \\ I_{w_2} & 0 \end{pmatrix}$
$UR(e_{(k,l),4}^{n-1})$	$S_{2,4}^{ver} = \begin{pmatrix} 0 & 0 \\ I_{h_4} & 0 \end{pmatrix}$	$S_{2,4}^{hor} = \begin{pmatrix} 0 & 0 \\ I_{w_4} & 0 \end{pmatrix}$
$r_3^{n-1,DR} : DCT(r_3^{n-1,DR}) = S_{3,3}^{ver} E_{(k,l),3}^{n-1} S_{3,3}^{hor} + S_{3,4}^{ver} E_{(k,l),4}^{n-1} S_{3,4}^{hor}$		
$LR(e_{(k,l),3}^{n-1})$	$S_{3,3}^{ver} = \begin{pmatrix} 0 & I_{h_3} \\ 0 & 0 \end{pmatrix}$	$S_{3,3}^{hor} = \begin{pmatrix} 0 & 0 \\ I_{w_3} & 0 \end{pmatrix}$
$LL(e_{(k,l),4}^{n-1})$	$S_{3,4}^{ver} = \begin{pmatrix} 0 & I_{h_4} \\ 0 & 0 \end{pmatrix}$	$S_{3,4}^{hor} = \begin{pmatrix} 0 & I_{w_4} \\ 0 & 0 \end{pmatrix}$
$r_4^{n-1,DR} : DCT(r_4^{n-1,DR}) = S_{4,4}^{ver} E_{(k,l),4}^{n-1} S_{4,4}^{hor}$		
$LR(e_{(k,l),4}^{n-1})$	$S_{4,4}^{ver} = \begin{pmatrix} 0 & I_{h_4} \\ 0 & 0 \end{pmatrix}$	$S_{4,4}^{hor} = \begin{pmatrix} 0 & 0 \\ I_{w_4} & 0 \end{pmatrix}$

$DCT(e_{(k,l),4}^n)$ can be easily extracted by performing inverse VLC and inverse quantization. These coefficients are useful for calculating $e_{(k,l),1}^{n,DR}$, $e_{(k,l),2}^{n,DR}$, $e_{(k,l),3}^{n,DR}$, and $e_{(k,l),4}^{n,DR}$. The representations of $e_{(k,l),1}^{n,DR}$, $e_{(k,l),2}^{n,DR}$, $e_{(k,l),3}^{n,DR}$, and $e_{(k,l),4}^{n,DR}$ are also shown in Fig. 10. In this figure, it is found that $e_{(k,l),1}^{n,DR}$ is equal to $e_{(k,l),1}^n$, i.e., $DCT(e_{(k,l),1}^{n,DR}) = DCT(e_{(k,l),1}^n)$. Besides, $e_{(k,l),2}^{n,DR}$, $e_{(k,l),3}^{n,DR}$, and $e_{(k,l),4}^{n,DR}$ are overlapping with $e_{(k,l),2}^n$, $e_{(k,l),3}^n$, and $e_{(k,l),4}^n$, respectively, without any shifting. Therefore, instead of using the shifting operator, a new chopping operator is necessary for extracting $e_{(k,l),i}^{n,DR}$ (for $i = 2, 3$, and 4) from $e_{(k,l),i}^n$. Given the motion vector $mv_{(k,l)}^n$ of $MB_{(k,l)}^n$, $(\Delta x, \Delta y)$, we can determine the necessary chopping region of each $e_{(k,l),i}^{n,DR}$ when $i = 2, 3$, and 4. For example, in Fig. 10, $w_2 = w_4 = 8 - \Delta x$, $h_3 = h_4 = 8 - \Delta y$, and $h_2 = w_3 = 8$. The derivation of the shifting operator can be also done in that of the chopping operator. Given $e_{(k,l),i}^n$ and the values of h_i and w_i , we can describe $e_{(k,l),i}^{n,DR}$ in the spatial domain through matrix multiplications again

$$e_{(k,l),i}^{n,DR} = c_{h_i}^{row} e_{(k,l),i}^n c_{w_i}^{col} \quad (25)$$

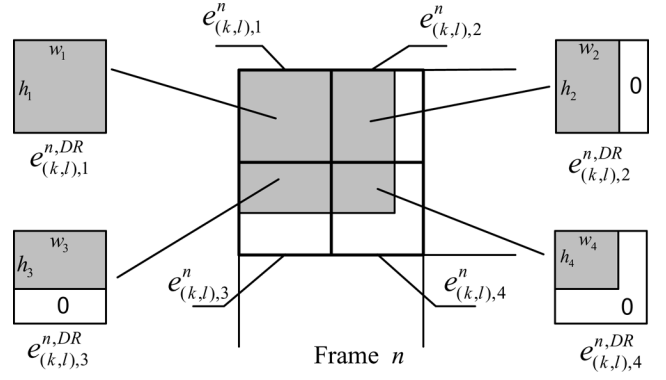


Fig. 10. Block level in frame n.

TABLE II
CHOPPING MATRICES $c_{h_i}^{row}$ AND $c_{w_i}^{col}$ FOR $e_{(k,l),i}^{n,DR}$ WHEN $i = 2, 3$, AND 4

$e_{(k,l),2}^{n,DR} : DCT(e_{(k,l),2}^{n,DR}) = C_{h_2}^{row} E_{(k,l),2}^n C_{w_2}^{col}$	
$c_{h_2}^{row} = I_8$	$c_{w_2}^{col} = \begin{pmatrix} 0 & 0 \\ 0 & I_{w_2} \end{pmatrix}$
$e_{(k,l),3}^{n,DR} : DCT(e_{(k,l),3}^{n,DR}) = C_{h_3}^{row} E_{(k,l),3}^n C_{w_3}^{col}$	
$c_{h_3}^{row} = \begin{pmatrix} I_{h_3} & 0 \\ 0 & 0 \end{pmatrix}$	$c_{w_3}^{col} = I_8$
$e_{(k,l),4}^{n,DR} : DCT(e_{(k,l),4}^{n,DR}) = C_{h_4}^{row} E_{(k,l),4}^n C_{w_4}^{col}$	
$c_{h_4}^{row} = \begin{pmatrix} I_{h_4} & 0 \\ 0 & 0 \end{pmatrix}$	$c_{w_4}^{col} = \begin{pmatrix} 0 & 0 \\ 0 & I_{w_4} \end{pmatrix}$

where $c_{h_i}^{row}$ and $c_{w_i}^{col}$ are the row and column of the chopping matrices, which are used to remove h_i rows and w_i columns in $e_{(k,l),i}^n$ and they are denoted by $\begin{pmatrix} I_{h_i} & 0 \\ 0 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & 0 \\ 0 & I_{w_i} \end{pmatrix}$, respectively.

In the example as shown in Fig. 10, the selected chopping matrices used for obtaining $e_{(k,l),1}^{n,DR}$, $e_{(k,l),2}^{n,DR}$, $e_{(k,l),3}^{n,DR}$, and $e_{(k,l),4}^{n,DR}$ are tabulated in Table II. By applying DCT to (25) and making use of distributive property in DCT, we obtain

$$DCT(e_{(k,l),i}^{n,DR}) = C_{h_i}^{row} E_{(k,l),i}^n C_{w_i}^{col} \quad (26)$$

where $C_{h_i}^{row} = DCT(c_{h_i}^{row})$ and $C_{w_i}^{col} = DCT(c_{w_i}^{col})$ are the chopping matrices in the DCT domain.

By adopting the shifting and chopping operators, $DCT(\hat{e}_{(k,l)}^{n,DR})$ in (20) can be obtained with reduced computational burden of the transcoder because it avoids the conversion processes operating back and forth between the DCT domain and the spatial domain.

In practical implementation, the DCT representations of the shifting and chopping matrices in Tables I and II can be pre-computed and stored in memory. From these tables, we can see that these matrices are in the form of $\begin{pmatrix} 0 & 0 \\ I_m & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & I_m \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} I_m & 0 \\ 0 & 0 \end{pmatrix}$,

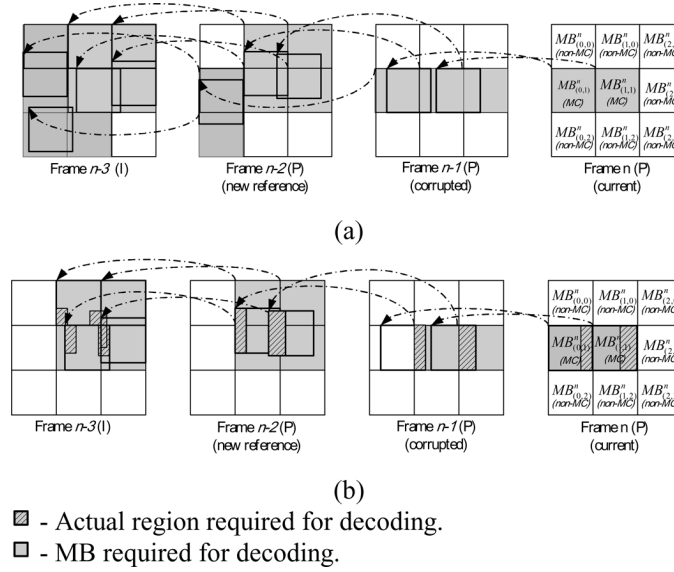


Fig. 11. Situation of MC MBs adopting our proposed DCT-domain techniques (a) without region tracking and (b) with region tracking.

or $\begin{pmatrix} 0 & 0 \\ 0 & I_m \end{pmatrix}$ where $m = 1, 2, \dots, 8$. There are 8 different matrices in each form. For $m = 8$, all four forms are the same. Therefore, 29×8 matrices need to be stored.

D. Region Tracking for Decoding Process

After obtaining $\text{DCT}(\hat{e}_{(k,l)}^{n,\overline{\text{DR}}})$, we now examine how the DCT coefficients of prediction errors in the $\overline{\text{DR}}$, $\text{DCT}(\hat{e}_{(k,l)}^{n,\overline{\text{DR}}})$, which is another necessary component of (20), can be calculated. In Fig. 2, the dominant motion vector $mv_{(k,l)}^{n-1}$ of $\text{MB}_{(k,l)}^{n-1}$ is used to re-compose the new motion vector $\hat{mv}_{(k,l)}^n$. But other MBs may have different motion vectors. If one of these motion vectors is different from the dominant motion vector, it is impossible to re-use the DCT coefficients in that MB. The spatial-domain re-encoding technique is, therefore, needed to compute $\text{DCT}(\hat{e}_{(k,l)}^{n,\overline{\text{DR}}})$. In other words, $\hat{e}_{(k,l)}^{n,\overline{\text{DR}}}$ is obtained by subtracting the pixel intensities of the reconstructed MB in frame n with the corresponding best-matched MB in the new reference, frame $n-2$. That is

$$\hat{e}_{(k,l)}^{n,\overline{\text{DR}}} = \text{MB}_{(k,l)}^{n,\overline{\text{DR}}} - \text{MB}_{(k,l)}^{n-2,\overline{\text{DR}}}. \quad (27)$$

For the pixel-domain re-encoding technique, all MBs from the previously nearest I-frame to frame n should be decoded. Then re-encoding is performed for those MC MBs with frame $n-2$ as the reference. With the help of various DCT-domain techniques, we suggest using a region tracking technique to trace only the indispensable MBs during the decoding and re-encoding processes. This technique traces from the current frame to the previously nearest I-frame and only the necessary MBs related to $\overline{\text{DR}}$ need to be decoded. For illustration, an example for re-encoding MC MBs without the region tracking technique is depicted in Fig. 11(a). This figure shows a situation in which $\text{MB}_{(0,1)}^n$ and $\text{MB}_{(1,1)}^n$ are MC MBs in frame n , and their corresponding motion vectors are denoted by the dotted arrows. Frame $n-1$ is the corrupted frame and frame $n-2$ is the new reference. To re-encode $\text{MB}_{(0,1)}^n$ and $\text{MB}_{(1,1)}^n$, the transcoder

examines the motion vectors in the encoded bitstream and all the related MBs from frame $n-3$ to frame n should be decoded. Three MBs (the shaded MBs) in frame $n-1$ are required to act as references for performing motion compensation of $\text{MB}_{(0,1)}^n$ and $\text{MB}_{(1,1)}^n$. These MBs in frame $n-1$ further require to use their corresponding MBs in frame $n-2$. This process continues until it reaches the previously nearest I-frame. In this example, the transcoder needs to decode 19 MBs (2, 3, 6, and 8 MBs from frames $n, n-1, n-2$, and $n-3$ respectively) for $\text{MB}_{(0,1)}^n$ and $\text{MB}_{(1,1)}^n$ from frame $n-3$ to frame n . This situation gets worse when the current transmitted frame is far away from the previously nearest I-frame. However, by applying the region tracking technique, we find that only $\overline{\text{DR}}$ s of MC MBs needs to be re-encoded in the pixel domain. Therefore, some decoded MBs in Fig. 11(a) do not actually contribute to $\text{MB}_{(0,1)}^n$ and $\text{MB}_{(1,1)}^n$. The idea of the region tracking technique is to identify the MBs having actual contribution to $\overline{\text{DR}}$ s of MC MBs in the previous frame. Let us use Fig. 11(b) to give a clearer account of our idea. In Fig. 11(b), both $\text{MB}_{(0,1)}^n$ and $\text{MB}_{(1,1)}^n$ are divided into two regions—the DR and the $\overline{\text{DR}}$. The $\overline{\text{DR}}$ is denoted by the block filled with diagonal lines, which is also the actual region to be re-encoded in the pixel domain. Since the number of pixels in each $\overline{\text{DR}}$ is smaller than that in the whole MB, the number of MBs to be decoded in the previous frames can be reduced. In this example, only 2, 2, 4, and 4 MBs are required in frames $n, n-1, n-2$, and $n-3$, respectively. This means that the necessary MBs used to re-encode the current MC MBs can be saved considerably. The longer the length of GOP, the larger the savings of the proposed technique.

V. EXPERIMENTAL RESULTS

A large amount of experimental work has been conducted to evaluate the performances of the proposed techniques for error-resilient transcoding using RPS. Let us denote them as RPS-DA, RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, and MCMB+RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$. RPS-DA uses the DA technique in (12) on quantized DCT coefficients for non-MC MBs while RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ employs the modified

TABLE III
SUMMARY OF TOOLS USED IN RPS-DA, RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, AND MCMB + RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$

Scheme	Non-MC MBs		MC MBs
	DA (equation (12))	Modified quantizer-dequantizer pair (equation (18))	Shifting and Cropping operators with region tracking
RPS-DA	✓		
RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$		✓	
MCMB +RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$		✓	✓

TABLE IV
REQUIRED NUMBERS OF TMM OPERATIONS FOR
MCMB + RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$

Shifting operator	TMM operations
$r_1^{n-1,DR}$	4
$r_2^{n-1,DR}$	2
$r_3^{n-1,DR}$	2
$r_4^{n-1,DR}$	1
Chopping operator	TMM operations
$e_{(k,l),1}^{n,DR}$	0
$e_{(k,l),2}^{n,DR}$	1
$e_{(k,l),3}^{n,DR}$	1
$e_{(k,l),4}^{n,DR}$	1
Total TMM operations (equivalent to no. of MBs to be decoded)	12 (3)

quantizer-dequantizer pair in (18). For MCMB + RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, it further extends RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ by adopting the shifting and chopping operators, and the region tracking technique for creating the prediction errors of MC MBs in the DCT domain. The tools used in RPS-DA, RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, and MCMB + RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ are summarized in Table III. The results are compared with that obtained using the pixel-domain re-encoding technique [15]. All test sequences have a length of 140 frames. “Mother and Daughter,” “Salesman,” “Calendar,” and “Foreman” are in CIF format while “Football” and “Table Tennis” are in SIF format. “Carphone” is a typical videophone sequence in QCIF format. All sequences were encoded by the MPEG-4 encoder [27] at two different bitrates. For all test sequences, the frame-rate of the video bitstream was 30 frames/s and the GOP length was 15 with an I-P structure.

In order to measure the computational requirements, we introduce a cost that can directly reflect the computational burden of the transcoder associated with various techniques. It is reasonable to approximate the cost to the required number of MBs to be decoded and re-encoded in the transcoder. For the pixel-domain re-encoding technique, the transcoder

requires to decode all MBs in P-frames from the previously nearest I-frame to the current frame, and re-encode all the MBs in the current frame with a new reference. The major steps in decoding a single MB are 2-D inverse DCT (2-D IDCT), dequantization, and motion compensation operations while the re-encoding process involves 2-D DCT (2-D DCT), quantization, motion estimation, and compensation operations. The main difference in computational complexity between the decoding and re-encoding processes is the motion estimation operation. It depends on the motion estimation algorithm to be used for the re-encoding process. In general, measuring the difference between the costs of these two processes would not be an easy task. In spite of these, it is possible to assume that their complexities are the same for the sake of simplicity. To re-encode MC MBs in the current transmitted frame of the proposed RPS-DA and RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, only the relevant MBs in the previous frames to the re-encoded MC MBs should be decoded, as depicted in Fig. 11(b). In this situation, the number of necessary MBs to be decoded in the transcoder can be reduced significantly. For non-MC MBs, (12) and (18) signify that our RPS-DA and RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ require simple arithmetic operations instead of the complicated decoding and re-encoding processes. Note that the computational requirement of RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ is similar to that of the RPS-DA, and only two more addition and multiplication operations are needed. Practically, the multiplication by 2 in (18) can be implemented by a shift operator. Therefore, these arithmetic operations are negligible as compared with the 2-D DCT or 2-D IDCT operation. For MCMB + RPS- $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, the DCT-domain shifting and chopping operators are proposed to handle MC MBs. These operators involve some matrix manipulations. In general, a target block is predicted from up to four contributing blocks. The computation of DCT coefficients of contributing blocks actually requires pre- and post-multiplication of an 8×8 data block with two 8×8 matrices, where the matrices can be pre-processed and stored in the transcoder. Given an 8×8 data block D and two 8×8 matrices M_{pre} and M_{post} , the computation of $M_{\text{pre}}DM_{\text{post}}$ includes two matrix multiplications. Let us assume, for the sake of argument, that the type of $M_{\text{pre}}DM_{\text{post}}$ is called a TMM operation, which is taken as the unit to measure and compare the computational complexity of the proposed DCT-domain operators. Since the DCT-domain operations cover the majority of the computations of the whole transcoding process, using a TMM operation as the basic unit of computational complexity comparison can realistically reflect the actual efficiency of different algorithms.

TABLE V
SAVINGS OF USING THE PROPOSED MB-BASED TECHNIQUES AS COMPARED WITH THE PIXEL-DOMAIN RE-ENCODING TECHNIQUE IN RPS

Sequences	Bitrate (bits/s)	$RPS - DA$ or $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ (%)			$MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ (%)		
		RTD=1	RTD=2	RTD=3	RTD=1	RTD=2	RTD=3
Mother and daughter	537 kbps	78.5	77.5	76.9	91.2	91.5	86.2
	112 kbps	74.6	73.4	72.6	88.2	85.6	83.7
Salesman	1.14 Mbps	47.1	48.3	49.6	71.2	68.6	67.7
	256 kbps	57.7	58.4	59.1	80.0	76.8	76.4
Football	2.25 Mbps	42.4	42.4	42.6	62.8	59.3	56.9
	898 kbps	43.0	42.9	43.2	63.3	59.8	57.5
Calendar	1.64 Mbps	23.8	23.4	23.2	37.3	34.4	32.5
	828 kbps	23.1	22.6	22.4	37.5	33.7	31.7
Table Tennis	1.44 Mbps	31.2	30.5	30.1	43.1	40.2	38.3
	504 kbps	31.0	30.3	29.9	44.0	40.6	38.6
Foreman	1.12 Mbps	18.4	14.7	10.8	35.1	27.0	20.4
	354 kbps	17.0	13.6	10.2	33.9	25.2	19.1
Carphone	457 kbps	29.7	26.1	24.1	64.7	51.2	41.5
	102 kbps	29.0	25.2	23.2	64.9	52.2	43.1

For using the shifting operator, Table I shows that the computation of $DCT(r_1^{n-1,DR})$, $DCT(r_2^{n-1,DR})$, $DCT(r_3^{n-1,DR})$, and $DCT(r_4^{n-1,DR})$ requires 4 TMM operations, 2 TMM operations, 2 TMM operations, and 1 TMM operation, respectively. Therefore, 9 TMM operations are needed to obtain $DCT(r^{n-1,DR})$. Besides, from (26), the chopping operators require 3 TMM operations to compute all $DCT(e_{(k,l),2}^{n,DR})$, $DCT(e_{(k,l),3}^{n,DR})$, and $DCT(e_{(k,l),4}^{n,DR})$. Putting all of these together, 12 TMM operations are required to compute each $DCT(\hat{e}_{(k,l)}^n)$ for $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$. On the other hand, for using the pixel-domain re-encoding technique in RPS, the major computation is from 2-D IDCT. Since IDCT is a separable transform, its 2-D version can be expressed in a matrix form $x = C^T X C$, where X and x are the 8×8 matrices containing DCT coefficients and pixel values, respectively. C is the 8×8 transform kernel and C^T is the transpose of C . Therefore, 4 TMM operations are required for each 16×16 MB. Consequently, the computational requirement of the proposed DCT-domain operators for manipulating one MB is equal to three (12/4) times as that of decoding one MB using the pixel-domain re-encoding technique, as summarized in Table IV.

By taking all these considerations, a detailed comparison of the computational savings with different RTD of the proposed techniques is given in Table V. All data in Table V are the average savings when an error occurs in all possible frames. We show that all RPS-DA, $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, and $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ outperform the RPS scheme using the pixel-domain re-encoding technique in all sequences. For RPS-DA and $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, Table V shows that the results are more significant for “Mother and Daughter”

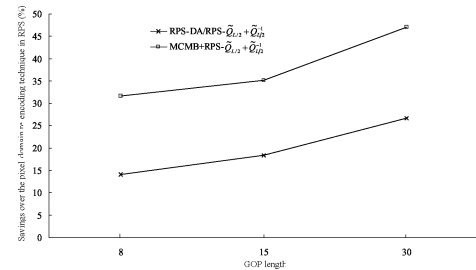


Fig. 12. Savings of various techniques for three different GOP lengths in the “Foreman” sequence encoded at 1.12 Mbits/s when RTD corresponds to the duration of encoding time for 1 frame.

and “Salesman” sequences. It is due to the reason that these sequences contain more non-MC MBs, in which the technique of DA or $\tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ can be performed more efficiently. For sequences containing high motion activities such as “Football,” “Calendar,” “Table Tennis,” “Foreman,” and “Carphone,” the savings of RPS-DA and $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ diminish. To further reduce the number of MBs to be decoded, the proposed DCT-domain operators can work with $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ for MC MBs. Table V also shows that $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ produces further savings, about 10%–35%, as compared with that of $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$. This can be explained by the benefits of the DCT-domain operators which can manipulate the prediction errors as much as possible in the compressed form. Besides, the region tracking technique can further minimize the number of MBs to be decoded.

Table V also shows the effects of different RTD on the proposed techniques. From this table, it can be shown that both RPS-DA and $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ can retain the saving as

TABLE VI
AVERAGE PSNR PERFORMANCES OF THE SUCCEEDING FRAMES AFTER THE CORRUPTED FRAME USING DIFFERENT TECHNIQUES IN RPS. Δ PSNR REPRESENTS A PSNR GAIN OVER THE RE-ENCODING TECHNIQUE (UNIT:DB)

Sequences	Bitrate (bits/s)	Re-encoding technique			RPS-DA (Δ PSNR)			$RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ (Δ PSNR)			$MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ (Δ PSNR)		
		RTD=1	RTD=2	RTD=3	RTD=1	RTD=2	RTD=3	RTD=1	RTD=2	RTD=3	RTD=1	RTD=2	RTD=3
Mother & daughter	537k	44.65	44.61	44.68	44.80 (+0.15)	44.72 (+0.11)	44.73 (+0.05)	44.84 (+0.19)	44.78 (+0.17)	44.80 (+0.12)	44.88 (+0.23)	44.79 (+0.18)	44.79 (+0.11)
	112k	40.27	40.27	40.30	40.30 (+0.03)	40.29 (+0.02)	40.30 (+0.00)	40.35 (+0.08)	40.31 (+0.04)	40.32 (+0.02)	40.35 (+0.08)	40.31 (+0.04)	40.31 (+0.01)
Salesman	1.14M	42.84	42.79	42.68	42.98 (+0.14)	42.93 (+0.14)	42.68 (+0.00)	43.21 (+0.37)	43.04 (+0.25)	43.03 (+0.35)	43.28 (+0.44)	43.06 (+0.27)	42.98 (+0.30)
	256k	37.20	37.11	37.13	37.27 (+0.07)	37.16 (+0.05)	37.18 (+0.05)	37.30 (+0.10)	37.20 (+0.09)	37.24 (+0.11)	37.31 (+0.11)	37.22 (+0.11)	37.24 (+0.11)
Football	2.25M	41.67	41.62	41.48	41.89 (+0.22)	41.59 (-0.03)	41.30 (-0.18)	42.36 (+0.69)	42.25 (+0.63)	42.14 (+0.66)	42.37 (+0.70)	42.30 (+0.68)	42.17 (+0.69)
	898k	34.40	34.30	34.24	34.54 (+0.14)	34.4 (+0.1)	34.29 (+0.05)	34.64 (+0.24)	34.56 (+0.26)	34.52 (+0.28)	34.96 (+0.56)	34.65 (+0.35)	34.55 (+0.31)
Calendar	1.64M	41.49	41.46	41.42	41.72 (+0.23)	41.35 (-0.11)	40.97 (-0.45)	41.89 (+0.40)	41.79 (+0.33)	41.75 (+0.33)	41.89 (+0.40)	41.86 (+0.40)	41.84 (+0.42)
	828k	34.00	33.89	33.86	34.18 (+0.18)	33.91 (+0.02)	33.70 (-0.16)	34.28 (+0.28)	34.17 (+0.28)	34.15 (+0.29)	34.21 (+0.21)	34.14 (+0.25)	34.15 (+0.29)
Table Tennis	1.44M	42.01	41.78	41.69	42.68 (+0.67)	41.75 (-0.03)	41.62 (-0.07)	42.91 (+0.90)	42.77 (+0.99)	42.63 (+0.94)	42.98 (+0.97)	42.81 (+1.03)	42.67 (+0.98)
	504k	34.80	34.62	34.60	35.02 (+0.22)	34.79 (+0.17)	34.75 (+0.15)	35.06 (+0.26)	34.99 (+0.37)	34.93 (+0.33)	35.15 (+0.35)	34.99 (+0.37)	34.76 (+0.16)
Foreman	1.12M	41.93	41.84	41.83	42.63 (+0.70)	42.27 (+0.43)	42.17 (+0.34)	42.75 (+0.82)	42.54 (+0.70)	42.56 (+0.73)	42.92 (+0.99)	42.58 (+0.74)	42.53 (+0.70)
	354k	36.66	36.49	36.50	36.89 (+0.23)	36.69 (+0.20)	36.67 (+0.17)	36.94 (+0.28)	36.76 (+0.27)	36.79 (+0.29)	36.94 (+0.28)	36.76 (+0.27)	36.77 (+0.27)
Carphone	457k	43.17	42.44	42.26	43.88 (+0.71)	42.79 (+0.35)	42.34 (+0.08)	44.12 (+0.95)	43.13 (+0.69)	42.35 (+0.09)	44.15 (+0.98)	43.12 (+0.68)	42.38 (+0.12)
	102k	37.28	36.92	36.55	37.79 (+0.51)	37.17 (+0.25)	36.63 (+0.08)	37.86 (+0.58)	37.35 (+0.43)	36.63 (+0.08)	37.88 (+0.60)	37.38 (+0.46)	36.62 (+0.07)

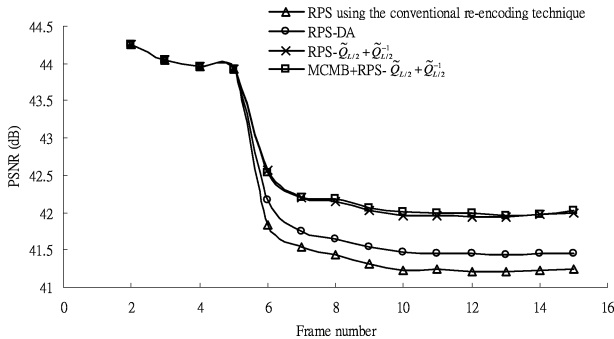


Fig. 13. PSNR performances of various techniques in the “Foreman” sequence encoded at 1.12 Mb/s when frame 2 is corrupted and RTD corresponds to the duration of encoding time for 3 frames.

RTD increases. On the other hand, it is noted that the saving of $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ drops as RTD increases. The reason behind is that the area of DR in the MC MB reduces for a longer RTD. Eventually, more MBs are required to undergo the pixel-domain process. However, significant amount of savings over the re-encoding technique can still be found when RTD is equal to 3, as shown in Table V.

We also demonstrate the savings of the proposed techniques over the pixel-domain re-encoding technique in the “Foreman” sequence encoded at 1.12 Mb/s when the length of GOP is varied, as shown in Fig. 12. For the pixel-domain technique, if the GOP length is long, more MBs need to be decoded, which

induces a significant increase of transcoding complexity. Fig. 12 shows that both of RPS-DA, $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, and $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ can provide more savings for long GOP length.

A detailed comparison of the average PSNR among different techniques for the succeeding frame right after the corrupted frame are tabulated in Table VI. It is found that the PSNR performance of the proposed RPS-DA is better than that of the re-encoding technique when the RTD corresponds to the duration of encoding time for one frame. When RTD is greater than 1, the average PSNR performance of RPS-DA drops, or it is even worse than the re-encoding technique in some sequences such as “Calendar,” “Football,” and “Table Tennis.” This is because the problem of the non-linear property of inverse quantization in RPS-DA becomes more significant for a longer RTD. This problem can be revolved using $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ and $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$. For different values of RTD, Table VI shows that the values of average PSNR achieved by $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ and $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ are very consistent. Furthermore, the quality degradation will not only be confined to the frame at the re-encoding point but can propagate and be accumulated in the subsequent P-frames. Such drift will last until the next I-frame. In Fig. 13, we have realized the effect of drift using the re-encoding technique and our proposed MB-based techniques for the “Foreman” sequences. It can be seen from Fig. 13 that the proposed RPS-DA, $RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$, and $MCMB + RPS - \tilde{Q}_{L/2} + \tilde{Q}_{L/2}^{-1}$ have remarkable PSNR improvements over the pixel-domain re-encoding technique. This

further demonstrates the effect of the proposed techniques when applying RPS to error-resilient transcoding.

VI. CONCLUSION

In this paper, we have addressed some issues on implementing error-resilient transcoding using the RPS. We have shown that a straightforward approach might result in much higher complexity of the transcoder and introduce re-encoding errors; these are not desirable. A new idea for the compressed-domain techniques applied to RPS, which can be adopted in an MPEG-4 encoded bitstream, has been proposed in this paper. The proposed techniques classify the MBs of the requested frame into two categories—non-MC MBs and MC MBs. Then it selects the necessary MBs adaptively, processes them in the compressed domain and sends the processed MBs to the decoder. For non-MC MBs, two techniques have been suggested. The first one is DA of quantized DCT coefficients to deactivate most of the complex modules of adopting RPS during transcoding. To avoid the non-linear problem of the DA technique, a modified quantizer-dequantizer pair for non-MC MBs has been proposed to maintain the reconstructed quality of the transcoded sequence. Besides, we have designed two DCT-domain operators for shifting and chopping the DCT coefficients of the encoded bitstream in order to compute the DCT coefficients of the target block in MC MBs. Through the matrix manipulation, the shifting operator can obtain some related DCT coefficients from the coefficients of its four overlapping DCT blocks in the encoded bitstream while the chopping operator can extract a part of DCT coefficients from the existing block. Both operators are carried out in the DCT domain. With the help of FDVS, these operators can be operated as much as possible in the DCT form in order to keep the benefits of the DCT-domain manipulation. Although the DCT-domain operators can reduce the required computations for MC MBs in the transcoder, some regions are still operated in the spatial domain. To minimize the use of the spatial-domain re-encoding technique, a region tracking technique has been employed to trace only the essential MBs during the re-encoding process. It can ensure that only the necessary MBs related to regions, that cannot be manipulated by the DCT-domain operators, are decoded. Experimental results show that the proposed techniques can reduce transcoder complexity significantly as well as preserving the video quality. Note that our proposed techniques only require a little increase in memory usage as compared with the pixel-domain re-encoding technique. For the pixel-domain technique, one picture memory for holding the new reference frame is needed during error-resilient transcoding using RPS. Although our proposed techniques do not need to store all pixel values of the new reference frame due to the benefit of using the compressed-domain manipulation, the memory for the new reference frame is still allocated for the sake of easy implementation. Besides, extra $29 \times 8 \times 8$ matrices for shifting and cropping operations are necessary to be pre-computed and stored in memory. The total memory size of these pre-computed matrices is insignificant for practical implementation. It is thus believed that the results of the present work will certainly be

useful for implementing the RPS in a practical error-resilient transcoding system.

REFERENCES

- [1] L. Chiariglione, "The development of an integrated audiovisual coding standard: MPEG," *Proc. IEEE*, vol. 83, no. 2, pp. 151–157, Feb. 1995.
- [2] *Video Coding for Low Bitrate Communication*, ITU-T Recommendation H.263, May 1997.
- [3] *Information Technology—Coding of Audio-Visual Objects: Visual*, ISO/IEC 14496-2, 1998.
- [4] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.
- [5] T. D. C. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 14–24, Autumn/Fall 1994.
- [6] F. A. Tobagi, "Distance learning with digital video," *IEEE Trans. Multimedia*, vol. 2, no. 1, pp. 90–93, Spring 1995.
- [7] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction for MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, Dec. 1998.
- [8] 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.
- [9] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, pp. 18–29, Mar. 2003.
- [10] K. T. Fung, Y. L. Chan, and W. C. Siu, "Low-complexity and high-quality frame-skipping transcoder for continuous presence multipoint video conferencing," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 31–46, Feb. 2004.
- [11] I. Ahmad, X. Wei, Y. Shun, and Y. Q. Zhang, "Video transcoding: An overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, Oct. 2005.
- [12] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proc. IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.
- [13] A. Vetro, J. Xin, and H. Sun, "Error resilience video transcoding for wireless communications," *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 14–21, Aug. 2005.
- [14] G. Reyes, A. R. Reibman, S. F. Chuag, and J. C. I. Chuang, "Error-resilient transcoding for video over wireless channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1063–1074, Jun. 2000.
- [15] S. Dogan, A. Cellatoglu, M. Uyguroglu, A. H. Sadka, and A. M. Kondoz, "Error-resilient video transcoding for robust internetwork communications using GPRS," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 453–464, Jun. 2002.
- [16] H.-J. Chiou, Y.-R. Lee, and C.-W. Lin, "Error-resilient transcoding using adaptive intra refresh for video streaming," in *Proc. IEEE Int. Symp. Circuits and Systems*, Vancouver, BC, Canada, May 23–26, 2004, vol. 3, pp. 777–780.
- [17] H.-J. Chiou, Y.-R. Lee, and C.-W. Lin, "Content-aware error-resilient transcoding using prioritized intrarefresh for video streaming," *J. Vis. Commun. Image Represent.*, vol. 16, no. 4–5, pp. 563–588, Aug.–Oct. 2005.
- [18] C.-M. Chen, C.-W. Lin, and Y.-C. Chen, "Adaptive error-resilience transcoding using prioritized intra-refresh for video multicast over wireless networks," *J. Signal Process.: Image Commun.*, vol. 22, no. 3, pp. 277–297, Mar. 2007.
- [19] S. Eminsoy, S. Dogan, and A. M. Kondoz, "Transcoding-based error-resilient video adaptation for 3G wireless networks," *EURASIP J. Adv. Signal Process.*, vol. 2007, p. 13, 2007, doi:10.1155/2007/39586, Article ID 39586.
- [20] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87, no. 10, pp. 1707–1723, Oct. 1999.
- [21] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," in *Proc. IEEE Global Telecommunications Conf.*, Nov. 1996, vol. 3, pp. 1503–1508.
- [22] N. Farber, B. Girod, and J. Villasenor, "Extensions of ITU-T recommendation H.324 for error-resilient video transmission," *IEEE Commun. Mag.*, vol. 36, no. 6, pp. 1201–128, Jun. 1998.
- [23] 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.

- [24] J. Youn, M. T. Sun, and C. W. Lin, "Motion estimation for high performance transcoding," *IEEE Trans. Consum. Electron.*, vol. 44, no. 8, pp. 649–658, Aug. 1998.
- [25] S. F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 1, pp. 1–11, Jan. 1995.
- [26] N. Merhav and V. Bhaskaran, "A fast algorithm for DCT-domain inverse motion compensation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1996, pp. 2307–2310.
- [27] *MPEG-4 Video Verification Model Version 18.0*, ISO/IEC JTC1/SC29/WG11 N3908, 2001.



Yui-Lam Chan (S'94–A'97–M'00) received the B.Eng. degree with first class honours and the Ph.D. degree from the Hong Kong Polytechnic University, Kowloon, in 1993 and 1997, respectively.

He joined the Hong Kong Polytechnic University in 1997, where he is now an Assistant Professor in the Department of Electronic and Information Engineering. He received the Faculty Merit Award in Teaching (Team) in 2005 and the Faculty of Engineering Research Grant Achievement Award in 2005. He was also actively involved in professional

activities. In particular, he was a reviewer, session chairman, and organizing/technical committee member of many international conferences. He was the Registration Chair and Session Chair of the 2004 International Symposium on Intelligent Multimedia, Video, and Speech Processing (ISIMP 2004) held in Hong Kong between October 20 and 22, 2004. He has published over 40 research papers in various international journals and conferences. His research interests include multimedia technologies, signal processing, image and video compression, video streaming, video transcoding, video conferencing, digital TV, error-resilient coding, and digital VCR.

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



Hoi-Kin Cheung received the B.Sc. (Hons.) and M.Phil. degrees from the University of New South Wales, Australia, in 2002, and the Hong Kong Polytechnic University in 2007, respectively.

He is currently a R&D Engineer at the Signal Communications Limited (a member of TeleEye Group). His research interests include image and video technology, video compression, and digital video error resilience.



Wan-Chi Siu (S'77–M'77–SM'90) received the Associateship from The Hong Kong Polytechnic University, 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 & 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 of the Department of Electronic and Information Engineering since 1992. He was Head of Department of Electronic and Information Engineering Department and subsequently Dean of Engineering Faculty between 1994 and 2002. He is currently the Director of the Centre for Multimedia Signal Processing at the same university. He has published 300 research papers, over 140 of which appeared in international journals, such as the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and he is an editor of the book *Multimedia Information Retrieval and Management* (Springer, 2003). His research interests include DSP, fast algorithms, transforms, wavelets, image and video coding, and computational aspects of pattern recognition.

Prof. Siu was a Guest Editor, Associate Editor, and Editorial Board Member of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, *Pattern Recognition*, *Journal of VLSI Signal Processing Systems for Signal, Image, Video Technology*, and the *EURASIP Journal on Applied Signal Processing*, in addition to other journals. He has been a keynote/invited speaker of some international conferences [such as the IEEE CPM'2002 (keynote speaker, Taipei, Taiwan, R.O.C.), IEEE ISIMP'04 (keynote speaker, Hong Kong), IEEE ICICS'07 (invited speaker, Singapore), and IEEE ICNNSP'08, Zhenjiang, China], and the organizer of many international conferences, including recently the MMSP'08 (Australia) as General Co-Chair, and three IEEE Society sponsored flagship conferences: ISCAS 1997 as Technical Program Chair; ICASSP 2003 as the General Chair; and ICIP 2010 (to be held in Hong Kong) as the General Chair. 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 Distinguished Presenter Award (1997), IEEE Third Millennium Medal (2000), the Best Teacher Award (2003), the Outstanding Award in Research (2003), Plaque for Exceptional Leadership from IEEE SPCB (2003), and the Honorable Mention Winner Award from Pattern Recognition (2004).