On Transcoding a B-Frame to a P-Frame in the Compressed Domain

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

***Abstract*—Only a limited number of methods have been proposed to realize heterogeneous transcoding, for example from MPEG-2 to H.263, or from H.264 to H.263. The major difficulties of transcoding a B-picture to a P-picture are that the incoming discrete cosine transform (DCT) coefficients of the B-frame are prediction errors arising from both forward and backward predictions, whilst the prediction errors in the DCT domain arising from the prediction using the previous frame alone are not available. The required new prediction errors need to be re-estimated in the pixel domain. This process involves highly complex computation and introduces re-encoding errors. We propose a new approach to convert a B-picture into a P-picture by making use of some properties of motion compensation in the DCT domain and the direct addition of DCT coefficients. We derive a set of equations and formulate the problem of how to obtain the DCT coefficients. One difficulty is that the last P-frame inside a GOP with an IBBP structure, for example, needs to be transcoded to become the last P-frame in the IPPP structure, and it has to be linked to the previous reconstructed P-frame instead of to the I-frame. We increased the speed of the transcoding process by making use of the motion activity which is expressed in terms of the correlation between pictures. The whole transcoding process is done in the transform domain, hence re-encoding errors are completely avoided. Results from our experimental work show that the proposed video transcoder not only achieves a speed-up of two to six times that of the conventional video transcoder, but it also substantially improves the quality of the video.**

***Index Terms*—B-picture and P-picture, compressed domain processing, heterogeneous transcoding, video coding and transcoding.**

I. Introduction

HOMOGENEOUS transcoding techniques of MPEG-2 to MPEG-2, H.261 to H.261, and H.263 to H.263 have been investigated [1]–[5]. However, there are also needs for heterogeneous transcoding, for example decoders of one format (e.g., H.263) must be able to receive videos which were previously encoded in another format (e.g., MPEG-2). This is particularly important for transmitting videos over low bandwidth channels or in hostile environments such as mobile networks and the Internet [6]. The emergence of the forthcoming Universal Mobile Telecommunication System (UMTS) that carries video, voice, and data is a good example. In this case, in contrast to homogeneous transcoding, picture type, directionality of motion vectors, and picture rate might all change. To resolve this problem, one straightforward approach for implementing heterogeneous transcoding is to cascade a decoder and an encoder. This is commonly known as pixel-domain transcoding and is shown in Fig. 1. The incoming MPEG-2 video bitstream is decoded in the pixel domain, and the decoded video frame is re-encoded by an H.263 encoder at the desired output bitrate according to the capability of the clients' devices and the bandwidth available on the network. This involves highly complex processing, large memory and long delays. Some recently proposed fast algorithms make use of the information from the incoming bitstream [7]–[15] to reduce the computational complexity. For example, motion vectors extracted from the incoming bitstream, after decoding, can be used to reduce significantly the complexity of transcoding, since motion re-estimation can be avoided [16]–[25]. However, the video needs to be decoded fully in the pixel domain and re-encoded according to the H.263 in this example. This pixel-domain transcoding approach suffers from its intrinsic double-encoding process, which introduces additional degradation and highly complex computations.

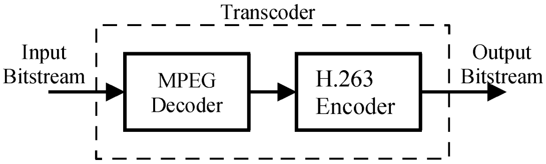

Fig. 1. Conventional heterogeneous transcoder.

Recently, some discrete cosine transform (DCT) domain transcoding processes have been introduced [26]–[30]. In these approaches, the incoming video bitstream is partially decoded in the DCT domain. Then, operations such as frame skipping, requantization, and video downscaling can be performed in this domain. The approach can provide an efficient method of bitrate reduction with low computational complexity. However, these are mainly homogeneous transcoding algorithms, while the formats of the incoming videos are out of our control. In order to support different kinds of video devices and video formats, heterogeneous transcoding becomes important, for mobile video applications for example. This includes mode decisions on transcoding between old and new coding modes, and new coding options to facilitate the transcoded videos with the best video quality in environments which contain various constraints. It is also interesting to note that other heterogeneous transcoding areas for mobile video and video

Manuscript received March 8, 2006; revised March 31, 2007. This work was supported by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University, and by the Research Grant Council of the Hong Kong SAR Government (PolyU 5234/03E). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yo-Sung Ho.

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: enwcsiu@polyu.edu.hk; enylchan@polyu.edu.hk; enktfung@eieserver.eie.polyu.edu.hk).

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

Digital Object Identifier 10.1109/TMM.2007.902895

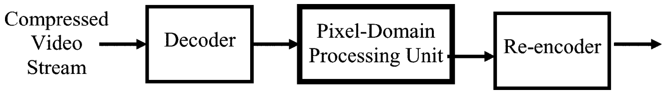

Fig. 2. Conventional cascaded pixel domain transcoding architecture.

browsing applications, for example, include P-frame to I-frame conversions, DCT-based video coding to wavelet-based video coding, H.263/MPEG to H.264/AVC conversion, etc.

Re-encoding errors are introduced during the re-encoding process. This problem arises from the fact that in order to transcode using a conventional transcoder, we have to decode the compressed video to obtain the pixel-domain signal and then re-encode it to form the signal with the desired format in the encoded domain as shown in Fig. 2. Decoding a compressed video does not mean that we are able to remove the original encoding errors which were mainly due to quantization. But in the re-encoding part, we will introduce further encoding errors again. These are referred to as re-encoding errors, which are additional errors introduced in the bit stream, due mainly to re-quantization.

The problem comes from the fact that no original (raw) video frames are available in the transcoder. In other words, only the previously encoded frames can be used inside the transcoder. When we transcode the video into a different format, we can only make use of this encoded bitstream instead of the original video sequence. Therefore, re-encoding errors are introduced during the transcoding process. In this work, we are trying to minimize the introduction of errors and to reuse directly and, as much as possible, the information obtained from the incoming bitstream.

Hence, the main objective of this paper is to resolve some of these bottlenecks and to propose possible solutions to alleviate the problems. The major difficulty is that when a B-picture is transcoded to a P-picture, the DCT coefficients which are computed with the residual errors between the predicted and original frames are not available from the incoming bitstream. Furthermore, the new P-picture is used immediately as a reference by the incoming frame. In other words, the next incoming P-picture needs to point to the previous frame instead of the original reference frame. In the next section, we will analyze this situation in detail. As a consequence, a compressed-domain heterogeneous transcoding will be derived, in which the transcoding process is carried out in the DCT domain where complete decoding and re-encoding are not required. This significantly reduces the processing complexity and preserves the quality of the transcoded video. In order to further improve the speed of realization a fast DCT-domain video transcoding, which uses the correlations between pictures that reflect the motion activities, is also proposed. The work of this paper introduces several novel formulations which represent some fundamental research concepts in heterogeneous transcoding. With further modifications of this techniques, the approach can be used for other applications, such as bitrate reduction transcoding (for converting a P-frame to a B-frame in order to save bits) and frame dropping (P to P conversion due to frame dropping in IPPPP…IPP format) in digital video recording applications.

The organization of this paper is as follows. Section II presents the formulation of the proposed DCT-based heterogeneous video transcoder, architecture and fast realization approaches. Experimental results are then given in Section III. Finally, concluding remarks are provided in Section IV.

II. DCT-Based Heterogeneous Video Transcoder

In [11], a heterogeneous transcoder is proposed for converting a B picture to a P picture. However, the major concern is that the transcoded DCT coefficients need to be recomputed in the pixel domain. In other words, re-encoding errors and highly complex computation are used during the transcoding process. In this paper, we present a new DCT-based heterogeneous video transcoding architecture. The new architecture focuses on the following areas.

2.1 to transcode a B frame to a P frame by re-estimating the DCT coefficients and motion vectors in the DCT domain using direct addition of DCT coefficients and motion compensation in the DCT domain (MCDCT).

2.2 to design a fast algorithm for transcoding the incoming P frame and make it refer to the previously transcoded B frame using the approach of backward subtraction of DCT coefficients.

2.3 to improve the speed of realization further by transcoding the B frame using uniform motion assumptions and checking.

The architecture of the proposed transcoder is shown in Fig. 3. 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. This is a DCT-based heterogeneous video transcoder which aims to transcode a video sequence from the MPEG-2 format to the H.261 or H.263 format. Thus, it is desirable to have efficient algorithms for the transcoding among I-pictures, P-pictures and B-pictures. Recall that for video codecs, a variety of coding formats are available, and hence many transcoding requirements exist. In this paper, we concentrate on converting a B-frame into a P-frame. Our approach can easily be generalized and is useful for most of these standards. For the sake of simplicity our discussion is confined to a simple case. Let us assume that a group of pictures (GOP) in the incoming bitstream has a length of eight pictures ($N = 8$); the distance between the anchor I/P pictures is set to 4 (i.e., $M = 4$, four pictures); and the format of the output picture sequence is either H.261 or H.263, with the sequence structures ($N = 8$, $M = 1$); as depicted in Fig. 4. The superscript, $'$, is used to denote the picture after performing the transcoding. In Fig. 4, pictures are presented in the display order, but are numbered in the encoding order. These two scenarios are the most probable conversion cases and the method can be generalized easily to other picture formats of the incoming and outgoing bitstreams. The following subsections describe our approach to transcoding the incoming bitstream in the DCT domain, the functions of the switches, and the advantages of the DCT-domain buffer arrangement, together with the details of other methods.

A. *Re-Estimate the DCT Coefficients and Motion Vectors in the DCT Domain*

Let us consider converting the first B-picture in a subgroup of an input bitstream, B_2, into a P-picture, P_1' which is predicted

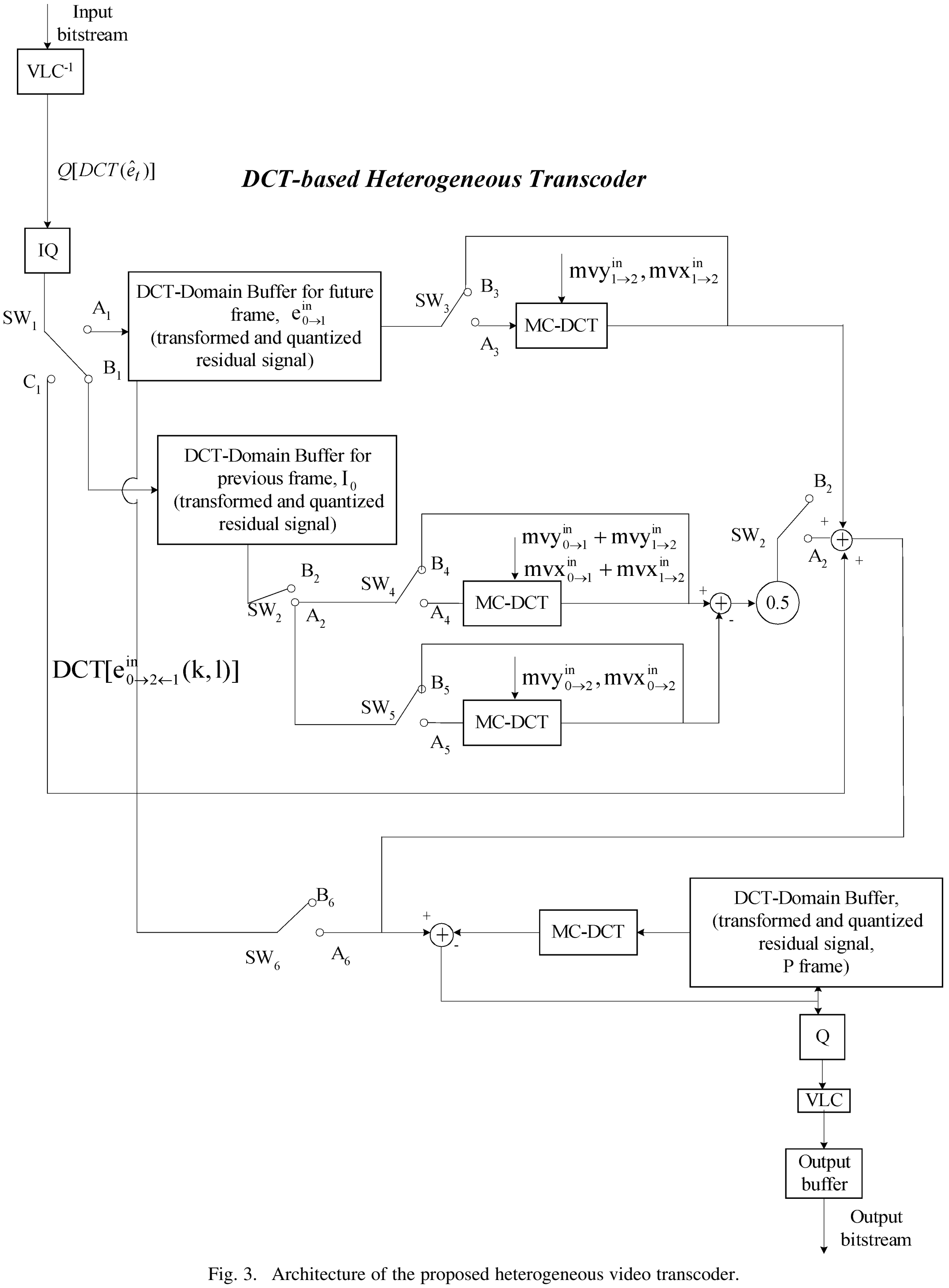
Input bitstream
VLC^{-1}
$Q[DCT(\hat{e}_t)]$
DCT-based Heterogeneous Transcoder
IQ
SW$_1$
A$_1$
B$_1$
C$_1$
DCT-Domain Buffer for future frame, $e^{in}_{0\to1}$ (transformed and quantized residual signal)
SW$_3$
B$_3$
A$_3$
$mvy^{in}_{1\to2}, mvx^{in}_{1\to2}$
MC-DCT
DCT-Domain Buffer for previous frame, I$_0$ (transformed and quantized residual signal)
B$_2$
SW$_2$
A$_2$
SW$_4$
B$_4$
A$_4$
$mvy^{in}_{0\to1} + mvy^{in}_{1\to2}$
$mvx^{in}_{0\to1} + mvx^{in}_{1\to2}$
MC-DCT
0.5
B$_2$
SW$_2$
A$_2$
$DCT[e^{in}_{0\to2\leftarrow1}(k,l)]$
SW$_5$
B$_5$
A$_5$
$mvy^{in}_{0\to2}, mvx^{in}_{0\to2}$
MC-DCT
B$_6$
SW$_6$
A$_6$
MC-DCT
DCT-Domain Buffer, (transformed and quantized residual signal, P frame)
Q
VLC
Output buffer
Output bitstream

Fig. 3. Architecture of the proposed heterogeneous video transcoder.

from frame I_0, as shown in Fig. 4. Our objectives are to find the new motion vector, $mv^{\text{out}}_{0\to1}$, for P'_1 with reference to I_0 and its associated residual error signal, $e^{\text{out}}_{0\to1}$, where the superscript "out" stands for the output motion vector of the transcoder and the subscript "$0 \to 1$" stands for frame 1 with reference to frame 0 in terms of the coding order. In order to save computation and reduce re-computational errors, we make full use of all available motion vectors and residual error signals of the original compressed video sequence.

Let us recall that the prediction error signal of the incoming B_2 frame is obtained from the previous I_0 frame and following P_1 frame, as shown in Fig. 5. It can be expressed in (1) as shown here:

$$\begin{aligned} & e^{in}_{0\to2\leftarrow1}(k+j, l+i) \\ &= \frac{B_2(k+j, l+i) - I_0(k+j+mvy^{in}_{0\to2}, l+i+mvx^{in}_{0\to2})}{2} \\ &+ \frac{B_2(k+j, l+i) - P_1(k+j+mvy^{in}_{1\to2}, l+i+mvx^{in}_{1\to2})}{2} \quad (1) \end{aligned}$$

where (k, l) represents the location of the upper left corner of a macroblock, (i, j) represents the spatial location within the

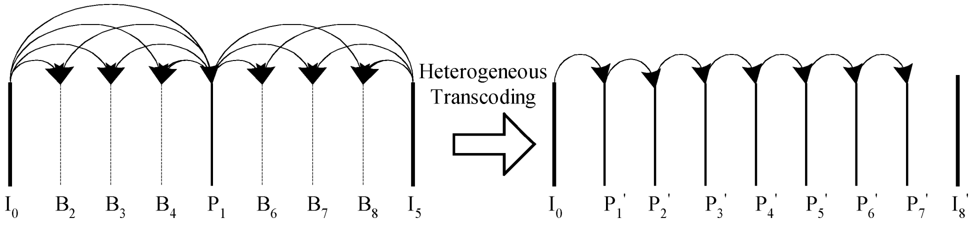

Fig. 4. Typical heterogeneous transcoding.

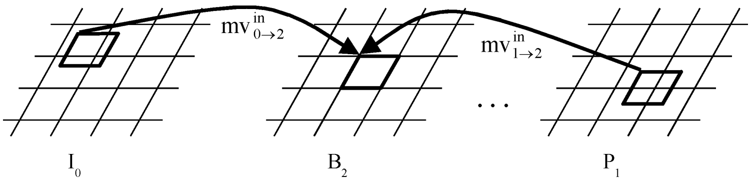

Fig. 5. Forward and backward motion vectors of B_2.

macroblock, mvx represents the x-displacement of motion vector mv, and mvy represents the y-displacement of motion vector mv. By applying the DCT to $e^{in}_{0\to2\leftarrow1}(k+j,l+i)$ and taking into account the linearity of the DCT, we obtain an expression for $e^{in}_{0\to2\leftarrow1}(k+j,l+i)$ in the DCT-domain

$$\begin{aligned}&2DCT[e^{in}_{0\to2\leftarrow1}(k,l)]\\&=DCT[B_2(k,l)-I_0(k+mvy^{in}_{0\to2},l+mvx^{in}_{0\to2})]\\&\quad+DCT[B_2(k,l)-P_1(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})].\end{aligned}\tag{2}$$

For the sake of simplicity, locations, say (i,j), of individual DCT coefficients are not included in the formulation. Our objective is to find the residual error $DCT[e^{\text{out}}_{0\to1}]$ of the output of the transcoder, which is expected to be obtained directly from the incoming bitstream, if possible. Let us write the quantized DCT coefficients of the output prediction errors $DCT[e^{\text{out}}_{0\to1}]$ of frame 1 as shown in the following equation:

$$\begin{aligned}&DCT[e^{\text{out}}_{0\to1}(k,l)]\\&\quad=DCT[B_2(k,l)-I_0(k+mvy^{\text{out}}_{0\to1},l+mvx^{\text{out}}_{0\to1})].\end{aligned}\tag{3}$$

This is an obvious formulation, since the output prediction error must be equal to the difference between frame $B_2(k,l)$ and the motion compensated $I_0(k+mvy^{\text{out}}_{0\to1},l+mvx^{\text{out}}_{0\to1})$ frame. For the rest of our formulation, it is desirable to make this output prediction error, $DCT[e^{\text{out}}_{0\to1}]$, between frame $I_0(k,l)$ and frame $P'_1(k,l)$ be in terms of the residual errors available in the original bit stream, including $e^{in}_{0\to2\leftarrow1}$ and $mv^{in}_{0\to1}$. In doing so, let us make $DCT[B_2(k,l)]$ the subject in (2) and substitute it into (3), we obtain

$$\begin{aligned}&2DCT[e^{\text{out}}_{0\to1}(k,l)]\\&=2DCT[B_2(k,l)]-2DCT[I_0(k+mvy^{\text{out}}_{0\to1},l+mvx^{\text{out}}_{0\to1})]\\&=2DCT[e^{in}_{0\to2\leftarrow1}(k,l)]+DCT[I_0(k+mvy^{in}_{0\to2},l+mvx^{in}_{0\to2})]\\&\quad+DCT[P_1(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})]\\&\quad-2DCT[I_0(k+mvy^{\text{out}}_{0\to1},l+mvx^{\text{out}}_{0\to1})].\end{aligned}\tag{4}$$

The output residual error is written in terms of $e^{in}_{0\to2\leftarrow1}$. Note that the new (output) motion vector, $mv^{\text{out}}_{0\to1}$, has not be defined yet. The above formulation can be simplified if we choose the incoming forward motion vector $mv^{in}_{0\to2}$ of frame B as the estimated new motion vector $mv^{\text{out}}_{0\to1}$ of the transcoded P frame, i.e.,

$$mv^{\text{out}}_{0\to1}=mv^{in}_{0\to2}.\tag{5}$$

This may not be the optimal choice for $mv^{\text{out}}_{0\to1}$, but it is close enough to the optimal motion vector obtained from the first principle. This choice decreases the volume of computation required to find the new motion vector, and simplifies the formulation of the output prediction errors $DCT[e^{\text{out}}_{0\to1}]$. Substituting (5) into (4), we have

$$\begin{aligned}2DCT[e^{\text{out}}_{0\to1}(k,l)]&=2DCT[e^{in}_{0\to2\leftarrow1}(k,l)]\\&\quad+DCT[P_1(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})]\\&\quad-DCT[I_0(k+mvy^{in}_{0\to2},l+mvx^{in}_{0\to2})].\end{aligned}\tag{6}$$

Equation (6) indicates that the new residual error depends upon the motion compensated version of $P_1(k,l)$. However, $P_1(k,l)$ was originally obtained with reference to $I_0(k,l)$ and with motion vector $mv^{in}_{0\to1}$, or

$$P_1(k,l)=I_0(k+mvy^{in}_{0\to1},l+mvx^{in}_{0\to1})+e^{in}_{0\to1}(k,l).$$

The second term of (6) is a shifted version of $P_1(k,l)$, or we may approximate the shifted version of the above equation shown as follows by using a linear model:

$$\begin{aligned}&P_1(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})\\&\cong I_0(k+mvy^{in}_{0\to1}+mvy^{in}_{1\to2},l+mvx^{in}_{0\to1},l+mvx^{in}_{1\to2})\\&\quad+e^{in}_{0\to1}(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})\end{aligned}$$

or by converting it into the DCT domain, we have

$$\begin{aligned}&DCT[P_1(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})]\\&\cong DCT[I_0(k+mvy^{in}_{0\to1}+mvy^{in}_{1\to2},l+mvx^{in}_{0\to1},l+mvx^{in}_{1\to2})]\\&\quad+DCT[e^{in}_{0\to1}(k+mvy^{in}_{1\to2},l+mvx^{in}_{1\to2})].\end{aligned}\tag{7}$$

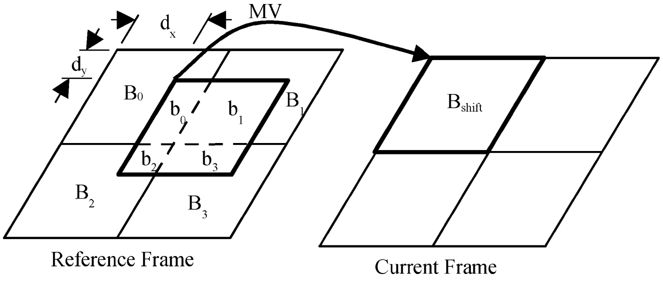

Fig. 6. Motion-compensated DCT coefficients for a block in the current frame.

Substituting (7) into (6), we can then make the new (output) prediction error also depend upon $e^{in}_{0\to 1}(k,l)$ in the DCT-domain as follows:

$$\begin{aligned}
&DCT[e^{\text{out}}_{0\to 1}(k,l)]\\
&\quad= DCT[e^{in}_{0\to 2\leftarrow 1}(k,l)]\\
&\quad+\frac{1}{2}DCT[e^{in}_{0\to 1}(k+mvy^{in}_{1\to 2}, l+mvx^{in}_{1\to 2})]\\
&\quad+\frac{1}{2}DCT[I_0(k+mvy^{in}_{0\to 1}+mvy^{in}_{1\to 2}, l+mvx^{in}_{0\to 1}+mvx^{in}_{1\to 2})]\\
&\quad-\frac{1}{2}DCT[I_0(k+mvy^{in}_{0\to 2}, l+mvx^{in}_{0\to 2})]. \qquad (8)
\end{aligned}$$

Equation (8) is good, since it relies only on the existing residual errors ($e^{in}_{0\to 2\leftarrow 1}$ and $e^{in}_{0\to 1}$) and frame $I_0(k,l)$. The first term can be obtained directly from the incoming bitstream. The second term is a motion-compensated (MC) prediction error of the incoming bitstream and it can be computed directly from the DCT-domain. By using the DCT-domain inverse motion-compensation proposed in [26], we can also obtain the third and fourth terms in the DCT-domain. Hence the new (output) motion vector, $mv^{\text{out}}_{0\to 1}$, can be obtained directly from the compressed domain.

A simple approach [30] to obtain the motion compensated DCT coefficients is to represent it as the sum of all horizontally and/or vertically displaced anchor blocks. Consider the motion vector for a shifted block, B_{shift}, in the current frame with motion vector (dy, dx) as shown in Fig. 6.

Due to the displacement (dy, dx), the motion-compensated block comes from four adjacent blocks, namely B_0, B_1, B_2 and B_3. The parts of the adjacent blocks forming the motion compensated block are labeled b_0, b_1, b_2 and b_3 as shown in Fig. 6. It is interesting to note that b_0 is the upper left corner of the required motion compensated block, but it comes from the lower right corner of block B_0 as shown in row 1 of Table I. Hence we have to shift subblock b_0 to the upper left corner using the displacement vector, (dy, dx). The shift is obtained by making use of the 8×8 shift operations $S_{RU}(s)$ and $S_{LD}(s)$, which are defined as

$$S_{RU}(s) = \begin{bmatrix} 0 & I_s \\ 0 & 0 \end{bmatrix} \qquad (9)$$

$$S_{LD}(s) = \begin{bmatrix} 0 & 0 \\ I_s & 0 \end{bmatrix} \qquad (10)$$

where s is the size of an identity matrix depending upon the value of (dy, dx), and I_s is the s by s identity matrix and $b_i = S_{iv}B_iS_{ih}$ with S_{iv} and S_{ih} defined in Table I.

TABLE I
OPERATIONS REQUIRED TO OBTAIN MOTION-COMPENSATED COEFFICIENTS

b_i (Sub-block)	Block Source	S_{iv}	S_{ih}
$b_{0,}$ Upper left corner	Lower right corner of B_0	$S_{0v} = S_{RU}(dy)$	$S_{0h} = S_{LD}(dx)$
$b_{1,}$ Upper right corner	Lower left corner of B_1	$S_{1v} = S_{RU}(dy)$	$S_{1h} = S_{RU}(8\text{-}dx)$
$b_{2,}$ Lower left corner	Upper right corner of B_2	$S_{2v} = S_{LD}(8\text{-}dy)$	$S_{2h} = S_{LD}(dx)$
$b_{3,}$ Lower right corner	Upper left corner of B_3	$S_{3v} = S_{LD}(8\text{-}dy)$	$S_{3h} = S_{RU}(8\text{-}dx)$

The sum of all b_i's gives us the motion compensated block, B_{shift}, or we can express it in matrix form as follows:

$$B_{\text{shift}} = \sum_{i=0}^{3} b_i = \sum_{i=0}^{3} S_{iv}B_iS_{ih} \qquad (11)$$

where S_{iv} is the vertical shift operation and S_{ih} is the horizontal shift operation as defined in Table I.

Let us relate the formulation between the pixel domain and DCT domain data by taking the DCT on both sides of (11)

$$\begin{aligned}
B'_{\text{shift}} &= DCT(B_{\text{shift}}) = DCT\left(\sum_{i=0}^{3}(S_{iv}B_iS_{ih})\right)\\
&= \sum_{i=0}^{3} DCT(S_{iv}B_iS_{ih}).
\end{aligned}$$

Recalling the linearity proper of the DCT, we have

$$DCT(A+B) = DCT(A) + DCT(B). \qquad (12)$$

Equation (12) is true for both 1-D and 2-D data. For videos, we have 2-D data, hence we can write,

$$DCT(AB) = DCT(A)DCT(B). \qquad (13)$$

Equation (13) says that the DCT of the product of A and B is equal to the matrix product of the DCT of A and the DCT of B. Making use of (12) and (13), we have

$$B'_{\text{shift}} = \sum_{i=0}^{3} DCT(S_{iv})DCT(B_i)DCT(S_{ih}). \qquad (14)$$

This equation says that the DCT of a motion compensated block can be obtained directly from the DCT domain with much simplified computational complexity. The DCT of S_{iv} and S_{ih} can be pre-computed because they are constants, whilst $DCT(B_i)$ is available from the incoming bitstream. Furthermore the transform errors are completely avoided. This formulation has great advantages compared with the pixel domain transcoder which requires calculation of the inverse DCT and then the DCT on the shifted data samples.

B. Fast Transcoding

Each P frame from the incoming bitstream refers to a previous I or P frame. It is also necessary to transcode the previous P frame because its previous B frame has been converted into

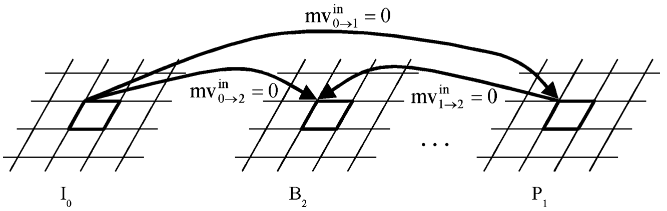

Fig. 7. Forward and backward motion vectors of B_2 for non-motion-compensated macroblocks.

a new P frame (see Fig. 4). In this situation, we can employ the dominant motion vector selection scheme and MC-DCT to transcode this P frame. For the non-motion compensated macroblock as shown in Fig. 7, we propose using the technique of direct subtraction of the DCT coefficients. Since the motion vectors are equal to zero in this case, we have

$$mv^{in}_{1\to2} = 0,\ mv^{in}_{0\to2} = 0\ and\ mv^{in}_{1\to0} = 0.$$

Using (2), we obtain

$$2DCT[e^{in}_{0\to2\leftarrow1}(k,l)] \\ = DCT[B_2(k,l)-P_1(k,l)]+DCT[B_2(k,l)-I_0(k,l)]. \quad (15)$$

Therefore, we have

$$DCT[B_2(k,l) - P_1(k,l)] \\ = 2DCT[e^{in}_{0\to2\leftarrow1}(k,l)] - DCT[B_2(k,l) - I_0(k,l)]. \quad (16)$$

In (16), the first term can be obtained directly from the incoming bitstream. The second term is derived from the previous subsection and it can be computed mainly on the DCT-domain. For the motion compensated macroblock, we can employ the direct subtraction of DCT coefficients, DCT-domain inverse motion-compensation proposed in [26] and dominant motion vector selection techniques to transcode the DCT coefficients.

C. Fast Transcoding of the B Frame Using A Uniform Motion Assumption and Checking

In the previous section, we successfully obtained the DCT coefficients from the DCT domain. A further simplification can be made by an approximation of the $DCT[e^{\rm out}_{0\to1}]$. This is done by assuming that motion activities between pictures are uniform [uniform motion assumption (UMA)], such that,

$$mv^{in}_{0\to2} = mv^{in}_{0\to1} + mv^{in}_{1\to2}. \quad (17)$$

Hence, (8) can be further simplified as

$$DCT[e^{\rm out}_{0\to1}(k,l)] = DCT[e^{in}_{0\to2\leftarrow1}(k,l)] \\ +\frac{1}{2}DCT[e^{in}_{0\to1}(k+mvy^{in}_{1\to2}, l+mvx^{in}_{1\to2})]. \quad (18)$$

In order to achieve high video quality for transcoding and low computational complexity, this assumption has to be verified.

TABLE II
SWITCH POSITIONS FOR DIFFERENT FRAME TYPES

Frame Type	*SW₁* Position
Previous frame	*B₁*
Future frame	*A₁*
B	*C₁*

TABLE III
SWITCH POSITIONS FOR USING UMAs

Uniform motion assumption (UMA)	*SW₂* Position
Yes	*B₂*
No	*A₂*

TABLE IV
SWITCH POSITIONS FOR CONVERTING DIFFERENT FRAME TYPES

Incoming B frame	*SW₆* Position
Yes	*B₆*
No	*A₆*

TABLE V
SWITCH POSITIONS FOR DIFFERENT CODING MODES

Coding mode	*SW₃* Position	*SW₄* Position	*SW₅* Position
Non MC	*B₃*	*B₄*	*B₅*
MC	*A₃*	*A₄*	*A₅*

A process, designated as uniform motion checking (UMC), is applied to check the validity of (17). If the motion activities between pictures cannot fulfill this requirement, SW_2 will be switched to position A_2 to estimate the new DCT coefficients using (8) instead of (18) (see Fig. 3).

In the next section, the use of UMA and UMC will be compared. Note that both approaches can obtain the targeted DCT coefficients in the DCT domain by reusing the incoming DCT coefficients in the above formulation. Using the above formulation, the architecture of the proposed transcoder is as shown in Fig. 3.

In Fig. 3, the video transcoder performs a variable-length decoding to extract the header information, coding mode, motion vectors and quantized DCT coefficients from the incoming bitstream. Switches SW_1, SW_2 and SW_6 are used to update the DCT-domain buffer for the transformed and quantized residual signal depending upon the coding mode originally used in the front encoder for the current macroblock being processed. SW_1 is used to update the DCT-domain buffer based on the frame type from the incoming bitstream. SW_2 is used to check the condition on UMC, and SW_6 is used to convert the incoming P frame to a newly reconstructed P frame since the reference frame is changed. Also, SW_3, SW_4 and SW_5 are used to control the MC-DCT modules. Functions of the switches are shown in Tables II–V. Note that as all operations are performed in the DCT-domain, a full re-encoding process is not required. Also, MC-DCT modules will be deactivated for non-MC macroblocks to speed up the transcoding process. Therefore, the computational complexity of the transcoder can be reduced significantly. In addition, degradation of the quality during the transcoding process can be avoided.

It is interesting to note that the effect of quantization does enter into the discussion of our transcoder. Actually, this is a key advantage of the formulations. In all the parts of the formulations, we have tried to form the new residual error signals (of the new P-frame) directly from the prediction errors of the original bit-stream (see the first paragraph of Section II-A). This can avoid decoding (inverse quantization and inverse DCT) and re-encoding (motion estimation, forward DCT and re-quantization) processes. Moreover, one significant point is that since in this case the original residual error signals in the original bit-stream are used, no re-quantization is required. This is because the residual error signals from the original bitstream have already been quantized. Hence the quantization is not involved in the whole of our transcoding procedure. This is the beauty of our formulations. The resultant effect is extremely important since it means that we are able to convert directly (without decoding and re-encoding) the residual error signals of the original bitstream into the required new residual error signals of the new P-frame. For the sake to clarity, the quantization is inserted in Fig. 3. However, neither inverse-quantization (IQ) nor the re-quantization (Q) processes are required in our realization, unless we want to have a further control of the bit-rate in the transcoder.

In this paper, we have investigated some novel techniques for converting a B-frame into a P-frame. Or, even more specifically we have laid down some basic formulations for an efficient conversion from two reference frames to a single reference frame. The conversion from H.261 to H.263 is used, since this is the simplest means, which can easily illustrate both the problem and the suggested solution. These basic formulations are extremely useful for the transcoding between MPEG-4 and H.264 (heterogeneous), or even for transcoding within the H.264 (homogeneous) video bit-streams [31], [32], say for example.

Let us describe two possible applications. 1) Due to the possible popularity of the H.264 coding approach, future videos may initially be encoded in the H.264 format, which in turn has to be transcoded into an H.263 bit-stream if we use some existing appliances (with the H.263 format) as the display device. 2) As a second example, our formulations are also extremely suitable for the reduction of reference frames due to a limitation in memory of the decoding device. Due the limitation in available memory, we sometime have to reduce the number of reference frames of a previously encoded video in the H.264 format, say for example, from five to three reference frames. In this case a block which originally refers to reference frames 3 and 4 must now refer to frame 3 only, which is similar to the situation discussed in this paper. Hence, direct transcoding within the transform domain is possible. This is able to avoid the complicated procedure of decoding and re-encoding; and will reduce computation and improve quality. Interested readers may investigate other situations similar to these examples. However, the study consists of a basket of other techniques, such as variable block size motion estimation, quarter pixel accuracy provision, arbitrary weights, etc., which could lead to a much wider scope of studies. This is a fruitful direction for further research.

III. Experimental Results

Extensive experiments have been performed to evaluate the overall efficiency of the proposed heterogeneous video transcoder. A fixed quantization parameter was used for all tested sequences, with I-B-P as the sequence structure (for $N = 3$ and $M = 1$). All B frames have to be converted into P frames inside the transcoder. These experiments evaluated the performances of the proposed techniques, including: 1) the DCT-domain transcoder with uniform motion assumption (DA+MCDCT+UMA) and 2) the DCT-domain transcoder with motion checking (DA+MCDCT+UMC). The front encoder was employed to encode video sequences with different spatial resolutions and motion characteristics. “Salesman”, “Foreman”, “Carphone” in QCIF (176×144) containing low motion activities and “Table Tennis” and “Football” with size 352×240 containing high motion activities were encoded by the MPEG-2 TM5 front encoder. For all testing sequences, the frame-rate of the incoming bitstream was 30 frames/s.

Careful experimental work has also been carried out to compare the performance of our architecture with the conventional cascaded pixel-domain transcoder (CPDT) as shown in Fig. 2. Detailed comparisons of the average PSNR between the CPDT and our proposed DCT-based transcoder using the uniform motion assumption are given in Table VI. Our proposed DCT-based transcoders outperform the CPDT in all cases. Note that this table also gives our experimental results on the bitrate variations. The bitrates of the originally encoded sequence, recoded by CPDT, and the bitrates using our approach are close. But our approach always requires slightly smaller bitrates compared with that of the CPDT. This is because the size of the residual errors may increase slightly after re-encoding, mainly due to re-quantization, in the re-encoding process of the CPDT; while our approach does not require re-encoding. It is interesting to note that for the new P-frame, basically one less motion vector is required for each macroblock. This is an innate bit saving property for converting a frame-B to a frame-P.

In terms of video quality, the results are more significant for sequences with high motion activities such as “Football” and “table Tennis”. This is because the proposed architecture transcodes the DCT coefficients in the DCT domain, which involves low computational complexity, and can reduce re-encoding errors. Significant improvement of 2.0–2.3 dB has been achieved compared with the conventional video transcoder. In terms of computational complexity, the speed up factor is about 4.5 to 7.3 times compared with the conventional video transcoder [7].

For sequences with low motion activities such as “Salesman” and “Carphone” sequences, the motion vectors are small. The MC-DCT module can be deactivated in these cases. Hence, significant improvement of about 5.6 to7.3 times can be achieved, as shown in Table VI. The speed up is the ratio measured by counting the number of operations required by the CPDT, over the number of operations required by the proposed approach. We also compared experimentally the computer run-time required for both approaches for the practical implementation of the sequences in Table VI. Both ways of comparison arrive at similar speed up ratios; however our approach is certainly much faster than that of the CPDT.

Table VII compares the average PSNR and the complexities of our proposed transcoders using uniform motion assumption DA+MCDCT+UMA, and our proposed transcoder using

TABLE VI
PERFORMANCE COMPARISON FOR ALL TRANSCODED B-FRAMES USING THE PROPOSED DCT-BASED HETEROGENEOUS TRANSCODER WITH UMA AND CONVENTIONAL CPDT

Sequences	Input bitrate	CPDT		DA+MCDCT+UMA			
		Average PSNR	Transcoded Bitrate	Average PSNR	Transcoded Bitrate	Speed-up ratio as compared with CPDT (number of operations for the proposed algorithm/ number of operations for the CPDT)	Speed-up ratio as compared with CPDT (time of the proposed algorithm / time of the CPDT)
Salesman (176x144)	64k	33.34	64.2k	35.61	64.1k	6.89	6.9
	128k	36.78	128.1k	39	128.1k	7.33	7.2
Foreman (176x144)	64k	30.53	64.5k	32.69	64.3k	4.65	4.5
	128k	34.28	128.4k	36.56	128.3k	4.85	4.7
Carphone (176x144)	64k	32.11	64.2k	34.15	64.2k	5.55	5.6
	128k	34.74	128.3k	36.93	128.3k	5.72	5.7
Table Tennis (352x240)	1.5M	32.41	1.53M	34.54	1.53M	5.06	5
	3M	35.04	3.05M	37.33	3.04M	5.30	5.2
Football (352x240)	1.5M	30.18	1.54M	32.32	1.53M	4.47	4.5
	3M	33.89	3.04M	36.20	3.04M	4.73	4.6

TABLE VII
COMPARISON OF THE PERFORMANCE OF ALL TRANSCODED B-FRAMES USING THE PROPOSED DCT-BASED HETEROGENEOUS TRANSCODER WITH UMA AND UMC

Sequences	Input bitrate	DA+MCDCT+UMA	DA+MCDCT+UMC	
		Average PSNR	Average PSNR	Speed-up ratio as compared with CPDT
Salesman (176x144)	64k	35.61	35.62	5.30
	128k	39	39.01	6.33
Foreman (176x144)	64k	32.69	33.22	2.32
	128k	36.56	37.1	2.49
Carphone (176x144)	64k	34.15	34.34	3.22
	128k	36.93	37.08	3.43
Table Tennis (352x240)	1.5M	34.54	34.78	2.69
	3M	37.33	37.54	2.94
Football (352x240)	1.5M	32.32	32.91	2.16
	3M	36.20	36.80	2.38

uniform motion checking named as DA+MCDCT+UMC. DA+MCDCT+UMA has similar performance with the DA+MCDCT+UMC in terms of the computational complexity. The quality can be further increased, therefore, DA+MCDCT+UMC is more suitable for high-quality video transcoding applications.

From Table VII, it can be seen that the PSNR performance of DA+MCDCT+UMC is better than that of the DA+MCDCT+UMA. This result is not unexpected since not all the motion vectors obey (17). For video sequences with low motion activities, (17) is satisfied for most macroblocks. Therefore, DA+MCDCT+UMA works well. However, it is beneficial to check this assumption to avoid unnecessary quality degradation especially when for transcoding video sequences containing high motion activities. In the "Salesman" and Carphone" sequences, the speed up performance is about 3.3 to 6.3, since most of the macroblocks are coded in non-MC mode. The average PSNR performance is very close to that of the case DA+MCDCT+UMA. For the "Football" and "Foreman" sequences, the average PSNR performance can be further improved by about 0.53-0.6 dB. The speed up performance is about 2.2 to 2.5 times.

In our scheme with uniform motion checking, estimation errors may occur since the DCT coefficients are obtained by using an approximation. The approximation is due to the assumption of linear motion activities, and the algorithm is usually able to reflect effectively what the motion is like in the real situation. Besides, if there is a sudden change in motion, the motion vector reuse approach and our proposed approach both need refinement in order to obtain an optimal motion vector. In fact, the proposed approach can minimize the re-encoding error effectively as shown in Fig. 8. Note that no I-frames propagate re-encoding errors, and in this example a picture format of IBBPBBPBBPBBI,IBB… was used. Fig. 8 shows the performance of the proposed transcoder (DA+MCDCT+UMC) compared with the conventional pixel domain transcoder (CPDT). Our experimental results also show that the uniform motion checking mechanism worked very well, which was able to reflect effectively the real situations. Note that even when there is a sudden motion change, the proposed approach can minimize the re-encoding error effectively. It is also clear from Fig. 8 that the proposed transcoder is able to have a 2-dB gain in terms of PSNR.

IV. CONCLUSION

In this paper, we have proposed a new architecture for a low complexity and high quality heterogeneous video transcoder which can effectively convert a B-picture into a P-picture. We derived a set of equations and formulated the problem of how to obtain the DCT coefficients. By using the motion vectors and the DCT coefficients from the incoming bitstream, the proposed video transcoder is able to directly transcode the B-picture into a P-picture in the DCT domain. In addition, a fast algorithm to transcode a B frame into a P frame using backward subtraction of DCT coefficients was proposed. The low degree of computational complexity was achieved by performing video transcoding in the DCT-domain using 1) techniques involving motion compensation in the DCT domain, and 2) an

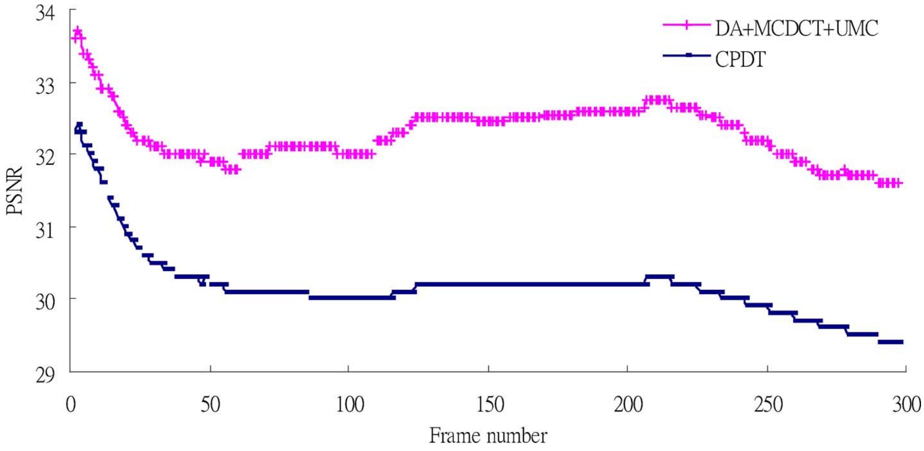

Fig. 8. PSNR performance of the proposed transcoder (DA+MCDCT+UMC) compared with the conventional transcoder (CPDT) for the foreman video sequence.

indirect addition of DCT coefficients to re-estimate DCT coefficients in the DCT domain. We proposed a fast algorithm to further speed up the transcoding process in the DCT domain using correlations which can reflect motion activities between pictures. Since the transcoding process was performed in the DCT domain, the re-encoding errors were significantly reduced. The overall performance of the proposed architecture produces a picture with a quality better than that of the conventional video transcoder. Experimental results confirm that the proposed video transcoder achieves a better performance than the conventional video transcoder in terms of both quality and complexity.

References

[1] Video Codec Test Model, TMN8 97, ITU-T/SG15.

[2] Video Coding for Low Bitrate Communication 1997, ITU-T Recommendation H.263.

[3] Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Video 1996, ISO/IEC 13818-2.

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

[5] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: An overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 793–804, Oct. 2005.

[6] T. Warabino, S. Ohta, D. Morikawa, M. Ohashi, H. Nakamura, H. Iwashita, and F. Watanabe, "Video transcoding proxy for 3G wireless mobile Internet access," *IEEE Commun. Mag.*, vol. 10, pp. 66–71, 2000.

[7] 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. 999.

[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] 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, pp. 929–936, Sep. 1999.

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

[11] 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.

[12] 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.

[13] J. Youn, M.-T. Sun, and C.-W. Lin, "Motion estimation for high performance transcoding," *IEEE Trans. Consumer Electron.*, vol. 44, no. 3, pp. 649–58, Aug. 1998.

[14] P. A. A. Assuncao and M. Ghanabari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 8, pp. 953–967, 1998.

[15] H. Sun, W. Kwok, and J. W. Zdepski, "Architecture for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 2, pp. 191–199, May 1996.

[16] 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.

[17] Y. L. Chan and W. C. Siu, "Edge oriented block motion estimation for video coding," *Proc. Inst. Elect. Eng., Vis., Image Signal Process.*, vol. 144, no. 3, pp. 136–144, Jun. 1997.

[18] Y. L. Chan and W. C. Siu, "On block motion estimation using a novel search strategy for an improved adaptive pixel decimation," *J. Vis. Commun. Image Repres.*, vol. 9, no. 2, pp. 139–154, Jun. 1998.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[24] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 6, pp. 849–866, Nov. 1998.

[25] 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.

[26] J. Song and B. L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 767–775, Aug. 2000.

[27] P. A. A. Assuncao and M. Ghanabari, "Fast computation of MC-DCT for video transcoding," *Electron. Lett.*, vol. 4, pp. 284–286, 1997.

[28] U. Koc and K. Ray, "DCT-based motion estimation," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 948–965, Jul. 1998.

[29] N. Merhav and V. Bhaskaran, "Fast algorithms for DCT-domain image down sampling and for inverse motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 3, pp. 468–476, Jun. 1997.

[30] 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.

[31] J. Ostermann, J. Bornmans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, 2004.

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

Wan-Chi Siu (S'77–M'77–SM'90) received the Associateship from The Hong Kong Polytechnic University (HKPU) in 1975, the M.Phil. degree from The Chinese University of Hong Kong in 1977, 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 HKPU 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 the 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 of the same university. He has published 300 research papers, over 130 of which appeared in international journals, such as IEEE TRANSACTIONS ON SIGNAL PROCESSING and is an editor of the book *Multimedia Information Retrieval and Management* (New York: 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 Editorial Board Member of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, *Pattern Recognition*, the *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 speaker of many international conferences, including IEEE PCM-2002 (Taiwan) and the IEEE 2003 ICNNSP (Nanjing, China). He has been General Chair or Technical Program Chair of many international conferences, including the IEEE Society sponsored flagship conferences such as ISCAS'1997, ICASSP'2003, and ICIP'2010 (to be held in Hong Kong). Between 1991 and 1995, he was a member of the Physical Sciences and Engineering Panel of the Research Grants Council (RGC), Hong Kong Government, and in 1994, he chaired the first Engineering and Information Technology Panel 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).

Yui-Lam Chan received the B.Eng. (with first-class honors) degree and the Ph.D. degree from the Hong Kong Polytechnic University (HKPU) in 1993 and 1997, respectively.

He joined the HKPU in 1997, and is now an Assistant Professor in the Department of Electronic and Information Engineering. 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 received the Faculty Merit Award in Teaching (Team) in 2005 and the Faculty of Engineering Research Grant Achievement Award in 2005. During his studies, he was 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. He has been actively involved in professional activities, including as 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.

Kai-Tat Fung received the B.Eng. degreee, M.Phil., degree and Ph.D. degrees from the Hong Kong Polytechnic University (HKPU) in 1998, 2001, and 2005, respectively.

He is currently a Postdoctoral Research Fellow of the Department of Electronic and Information Engineering at HKPU. His research interests include image and video technology, video transcoding, and multimedia applications, such as video conferencing, audio compression, and blind signal separation.