DCT-BASED VIDEO FRAME-SKIPPING TRANSCODER

Kai-Tat Fung and Wan-Chi Siu

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ABSTRACT

Due to high computational complexity and quality degradation introduced by conventional frame-skipping transcoders, a DCT-based video frame-skipping transcoder is proposed recently. However, the transcoding process of the motion compensated macroblocks in the DCT domain becomes the bottleneck since IDCT and DCT processes are required. In this paper, we propose a new architecture of the frame-skipping transcoder to reduce the computational complexity of motion compensated macroblocks in the frame-skipping process. The new architecture transcodes the dominant region of an MC macroblock in the DCT domain by making use of the DCT coefficients from the incoming bistream and some pre-computed shifting operators. By using our proposed shifted version of the dominant vector, the re-encoding errors introduced in the dominant region can be avoided. On the other hand, an adaptive transcoding architecture to transcode the boundary of MC marcoblocks and to perform error compensation of MC macroblocks boundary is also proposed. Experimental results show that, as compared to the conventional transcoder and DCT-based transocder, the new architecture for frame-skipping transcoder is more robust to noise, gives rise to fewer requantization errors, and requires simple computational complexity.

1. INTRODUCTION

Networked multimedia services such as multipoint video conferencing, video on demand and digital TV, are emerging [1-5]. Due to different demands on a matching of the video source to channel constraints and characteristics, it is important for a video server to provide quality support services to heterogeneous clients or transmission channels. It should have the capability of performing transcoding [3-4], which is regarded as a process of converting a previously compressed video bitstream into a lower bitrate bitstream without modifying its original structure.

One of general approaches for implementing transcoding is to cascade a decoder and an encoder [3-4], commonly known as pixel-domain transcoding. The incoming video bitstream is at first fully decoded in the pixel domain, and the decoded video frame is re-encoded at the desired output bitrate according to the capability of the clients' devices and the available bandwidth of the network for storage or transmission. This process involves a high computational complexity, large memory size, and long delay. Recently, some fast algorithms have been proposed by making use of the information from the incoming bitstream [4] to reduce the computational complexity. For example, motion vectors extracted from the incoming bitstream after decoding can be used to reduce the complexity of the transcoding significantly since motion re-estimation can be avoided. In addition, the video quality of the pixel-domain transcoding approach suffers from its intrinsic double-encoding process, which introduces additional degradation. The effect of re-encoding errors is depicted in Figure 1 where the "Salesman" sequence was transcoded at half of the incoming frame rate. This Figure shows that re-encoding errors lead to a significant drop in the picture quality of about 3.5dB on average.

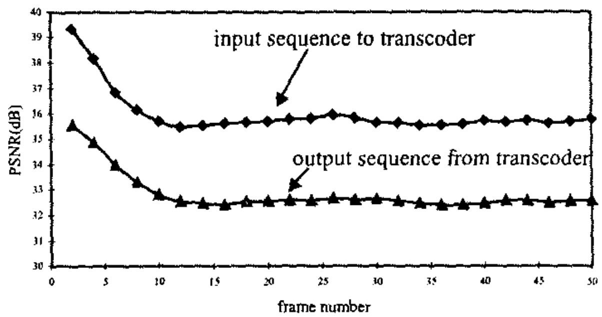


Figure 1: Quality degradation of conventional frame-skipping transcoder for the "Salesman" sequence. The peak signal-to-noise ratio (PSNR) of the frame-skipping pictures is plotted to compare with that of the same pictures which used directly a decoder without a transcoder.

2. PROPOSED DCT-BASED VIDEO FRAME-SKIPPING TRANSCODER

For a real image sequence, the block motion field is usually gentle, smooth, and varies slowly. As a consequence, the distribution of motion vectors is center-biased, as demonstrated by some typical examples as shown in Table 1, which gives the distribution of the coding modes for various sequences including "Salesman", "Foreman", "Carphone", "Table Tennis" and "Football". These sequences have been selected to emphasize different amount of motion activities. It is clear that about 76% and 27% of the macroblocks are coded without motion compensation for sequences containing low and high amount of motion activities respectively, for encoding with half-pixel precision. By using a direct addition of the DCT coefficients in the frame-skipping transcoder, the computational complexity and re-encoding errors can be reduced significantly for sequences containing more non-MC macroblocks. In this paper, we propose a new scheme to tackle the re-encoding error and high computational complexity introduced in the video transcoding process for MC macroblocks.

Table 1. Percentage of non-MC macroblock for various sequences using half-pixel accuracy.

Salesman	Foreman	Carphone	Table Tennis	Football
76%	42%	48%	54%	27%

In [5], we have proposed a direct addition of the DCT coefficients in frame-skipping transcoding to avoid re-encoding errors and to reduce the complexity for macroblocks coded without

0-7803-7761-3/03/$17.00 ©2003 IEEE

motion compensation. In this paper, we present a new frame-skipping transcoding architecture which is an extension of the work of [5]. The new architecture gives focus on the following areas.

1. Transcoding the dominant region of MC macroblocks in DCT domain

2. Using adaptive transcoding of MC macroblocks boundary

The architecture of the proposed transcoder is shown in Figure 4. The input bitstream is first parsed with a variable-length decoder to extract the header information, coding mode, motion vectors and quantized DCT coefficients for each macroblock. Let us define the incoming residual signal with quantization errors due to the front encoder as $\hat{e}_t = e_t + \Delta_t$ and its quantized DCT coefficients as $Q[DCT(\hat{e}_t)]$. Each macroblock is then manipulated independently. Two switches, SW_1 and SW_2, are employed to update the DCT-domain buffer, FB_{DCT}, for the transformed and quantized residual signal. The selection depends on the coding mode originally used in the front encoder for the current macroblock being processed. The switch positions for different coding modes and frame-skipping modes are shown in Table 2 and Table 3 respectively.

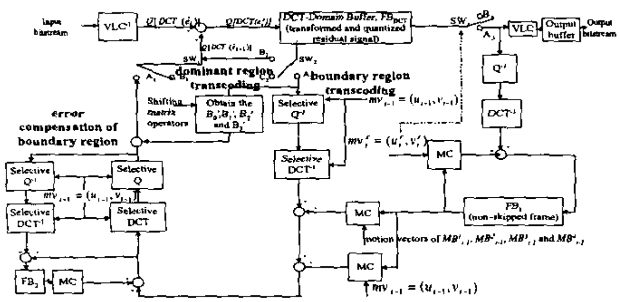


Figure 4. Architecture proposed for frame-skipping transcoder.

Table 2. Different coding modes of switches SW_1 and SW_2 of the proposed transcoder.

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

Table 3. Switch positions for different frame-skipping modes of the proposed transcoder.

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

2.1 Transcoding the dominant region of MC macroblocks in DCT domain

For MC macroblocks, direct addition cannot be employed since MB_{t-1} is not on a macroblock boundary, as depicted in Figure 5. In other words, $Q[DCT(\hat{e}_{t-1})]$ is not available from the incoming bitstream. Figure 5 also shows that MB_{t-1} is formed by using parts of four segments which come from its four neighboring macroblocks. Let us define these four neighboring macroblocks as MB^1_{t-1}, MB^2_{t-1}, MB^3_{t-1} and MB^4_{t-1}. It is possible to use the incoming quantized DCT coefficients of MB^1_{t-1}, MB^2_{t-1}, MB^3_{t-1} and MB^4_{t-1}, to come up with $Q[DCT(\hat{e}_{t-1})]$.

In this paper, we split the MC macroblocks into two regions: dominant region and boundary region as shown in Figure 6a. In the dominant region, we propose a shifted version of the dominant vector and a shifted operator to compute the DCT coefficients in the dominant region in DCT domain to achieve low computational complexity and avoid re-encoding errors.

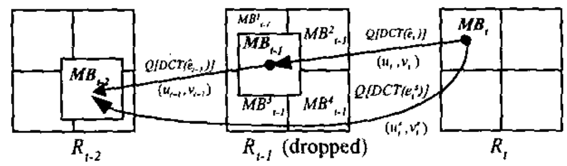
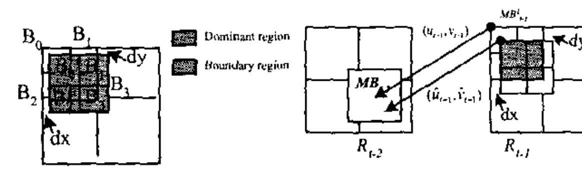


Figure 5. Residual signal re-computation of frame-skipping for MC macroblocks.

Figure 6b shows the scenario that the dominant macroblock is pointing at the previous non-skipped frame. The prediction errors obtained from the incoming bitstream are B_0, B_1,...B_{15} as shown in Figure 7a, where B_0, B_1, B_2, and B_3 have the motion vector which is considered as the dominant vector in this case, whilst all other blocks have three different motion vectors. Since this frame is skipped, the new prediction errors B_0', B_1', B_2' and B_3' of the previous non-skipped frame as shown in Figure 6a have to be available in order to employ the technique of direct addition of DCT coefficients. Since B_0, B_1, B_2 and B_3 have the dominant vector (u_{t-1}, v_{t-1})as shown in Figure 6b, the new prediction errors B_0', B_1', B_2' and B_3' can be obtained by using the shifted version of the dominant vector, $(\hat{u}_{t-1}, \hat{v}_{t-1})$, with a shift, (dx,dy), of (u_{t-1}, v_{t-1}). This is to avoid the re-encoding error in the dominant region if B_0', B_1', B_2' and B_3' are obtained directly from B_0, B_1, B_2, B_3, B_4, B_5, B_8, B_{10} and B_{12} in the DCT domain as shown in Figure 7a without performing requantization process during transcoding.



(6a) (6b)

Figure 6a) The dominant region and the boundary region of MB_{t-1}, 6b) Dominant macroblock and the previous non-skipped frame.

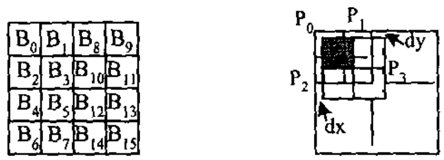
Since the shifted version of the dominant vector is used, the corresponding prediction error in the dominant region can be obtained without performing full re-encoding process. If the other three motion vectors have the same direction as the dominant vector, the whole new prediction (B_0', B_1', B_2' and B_3') can be obtained in the DCT domain. In other words, no re-encoding error will be introduced. Otherwise, a decomposition of the dominant region and boundary region is needed

The major idea to obtain the DCT coefficients in the dominant region is to represent it as the sum of all horizontally and/or vertically displaced anchor blocks. Then the DCT values of the B_0', B_1', B_2' and B_3' are constructed using the pre-computed DCT values of the shifting matrics.

Consider that P_0' is the skipped block of interest in pixel domain as shown in Figure 7b. P_0, P_1, P_2 and P_3 are its four neighboring blocks in pixel domain, and the motion vector is (dx,dy). The shaded regions in P_0, P_1, P_2 and P_3 are moved by an amount (dx,dy). Then P_0' can be represented by the following equation.

$$P_0' = \sum_{i=0}^{3} S_{i1} P_i S_{i2} \qquad (1)$$

where S_{ij}'s are matrics tabulated in Table 4.



(7a) (7b)

Figure 7a) Incoming DCT coefficients of four macroblocks, 7b) Dominant region and boundary region of MB_{t-1}

Table 4. The shifting operators.

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

Each I is an identity matrix of size hb=dy or hd=8-dy or wb=dx or wd=8-dx. As a pre-multiplication step, we have to shift the sub-block of interest horizontally while shift the sub-block vertically as a post-multiplication step. Four possible locations of the sub-block of interest: upper-left, upper-right, lower-left and lower-right are shown in Table 4.

From eq(1), we have,

$$DCT(P_0') = \sum_{i=0}^{3} DCT(S_{i1}) DCT(P_i) DCT(S_{i2}) \qquad (2)$$

This implies, $B_0' = \sum_{i=0}^{3} DCT(S_{i1})(B_i) DCT(S_{i2})$

Note that the DCT of S_{i1} and S_{i2} can be pre-computed and B_i can be extracted from the incoming bitstream. In this case the re-encoding errors are avoided and hence the computational complexity can be reduced. If the macroblocks MB^1_{t-1}, MB^2_{t-1}, MB^3_{t-1} and MB^4_{t-1} have the same motion vector as the dominant vector, B_1',B_2' and B_3' can be also obtained using this approach. Otherwise, a decomposition of the dominant region and boundary region is needed as shown in Figure 4.

Using the properties of DCT,

$$DCT(A+B)=DCT(A)+DCT(B) \qquad (3)$$

where A and B represent the pixels in dominant region and boundary region with size 8x8 respectively. We can split blocks B_1',B_2' and B_3'in two regions: dominant region and boundary region as shown in Figure 6a.

Then the dominant region of B_2' can be obtained by forcing the boundary blocks B_4 and B_5 to have zero DCT coefficients as shown in Figure 7a. Similarly, the dominant region of B_1' and B_3' can be obtained by forcing boundary blocks B_8 and B_{10} to have zero DCT coefficients and the boundary blocks B_5, B_{10} and B_{12} to have zero DCT coefficients as shown in Figure 7a. Hence the DCT value of the dominant region A can be obtained. Note that the DCT values of B_1', B_2' and B_3' can not be obtained since the DCT values in boundary region B, have not been considered as described in equation (3). In the following, the DCT values of boundary region B have to be obtained separately by using 1-D inverse DCT, motion compensation, forward 1D-DCT and requantization as shown in Figure 4. In this process, re-encoding error is introduced due to requantization.

2.2 Adaptive Transcoding of MC macroblocks boundary

For transcoding the MC boundary, only selective inverse quantization and selective inverse 1D-DCT of the quantized DCT coefficients of MB^1_{t-1}, MB^2_{t-1}, MB^3_{t-1} and MB^4_{t-1} (exclude the dominant one) need to be performed in the boundary as shown in Figure 4. When processing MB^1_{t-1}, MB^2_{t-1}, MB^3_{t-1} and MB^4_{t-1} (exclude the dominant one), only their corresponding *8×8* blocks, which have pixels overlapping with MB_{t-1} boundary region, are subject to the selective inverse 1D-DCT computation. Hence, the selective inverse 1D-DCT is only performed in the boundary, and the motion vector mv_{t-1} as shown in Figure 4 is needed as an input to the selective inverse 1D-DCT module to control which rows or columns are needed to perform the inverse 1D-DCT. In most cases, this approach significantly reduces the required number of columns or rows inverse DCT's as compared with the inverse 2D-DCT approach.

Each segment of the reconstructed pixels in MB_{t-1} boundary region can be obtained by adding its prediction errors to its motion-compensated segment of the previous non-skipped frame stored in FB_1, as shown in Figure 4. After all pixels in MB_{t-1} boundary region are reconstructed, we need to find the prediction error, $\hat{e}_{t-1}$. Actually, $\hat{e}_{t-1}$ is equal to the reconstructed pixel in MB_{t-1} boundary region subtracted from its corresponding MC macroblock from the previous non-skipped frame stored in FB_1, denoted as MB_{t-2} in Figure 5. In order to locate MB_{t-2}, we need to find a motion vector for MB_{t-1}. A shifted version of the dominant vector selection is used to select one dominant motion vector from four neighboring macroblocks with a shift (dx,dy). A dominant motion vector is defined as the motion vector carried by a dominant macroblock which is the macroblock that has the largest overlapped segment with MB_{t-1}.

Hence, $\hat{e}_{t-1}$ can be computed and it is selectively transformed by putting zero values in the dominant region and selectively quantized to $Q[DCT(\hat{e}_{t-1})]$. After the DCT coefficients of region B are obtained, the B_2', B_3'and B_1' can be reconstructed by adding DCT(A) and DCT(B) together using equation (3).

In Figure 5, the newly quantized DCT coefficient $Q[DCT(e^s_t)]$ of an MC macroblock can then be computed by adding $Q[DCT(\hat{e}_{t-1})]$ to the incoming $Q[DCT(\hat{e}_t)]$. Requantization is performed for the formation of $Q[DCT(\hat{e}_{t-1})]$ in MC macroblock boundaries, which will introduce additional re-encoding errors. Thus, a feedback loop is suggested as shown in Figure 4 to compensate for the re-encoding errors introduced in the previous frames. The selective forward and inverse 1D-DCT and selective quantization pairs in the feedback loop are mainly responsible for the minimization of re-encoding errors. For these MC macroblocks, the quantized DCT coefficients are inversely quantized. An inverse 1D-DCT is then performed to form $\hat{e}_{t-1}$ with a re-encoding error, which subtracts the original $\hat{e}_{t-1}$ to generate the re-encoding error, Δ^s_{t-1}. The re-encoding error is stored in FB_2, which can be written as

$$\Delta^s_{t-1} = DCT^{-1}\left(\left\lfloor \frac{DCT(\hat{e}_{t-1})}{q} \right\rfloor \times q\right) - \hat{e}_{t-1} \quad (4)$$

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

Since the motion vectors are highly correlated in the successive frames[6-7], it is observed that the spatial positions of MC macroblocks in certain frames are very close to the spatial positions of MC macroblocks in its subsequent frames. Thus, re-encoding errors stored in FB_2 are added to the prediction errors of MC macroblocks in the following P-frame to compensate for the re-encoding errors.

3. EXPERIMENTAL RESULTS

A large number of experiments have been performed to evaluate the overall efficiency of various frame-skipping transcoders. In the front encoder, the first frame of a sample sequence was encoded as intraframe (I-frame), and the remaining frames were encoded as interframes (P-frames). Note that the incoming bitstreams was encoded with half-pixel precision. The picture-coding modes was preserved during transcoding. The average PSNR performance of "Salesman", "Foreman" and "Carphone" sequences using various dynamic transcoder as shown in Table 5. Both DCT-based video transcoder using the techniques of direct addition of DCT coefficients (DA), forward motion vector selection (FDVS) and error compensation (EC) can achieve significant improvement in terms of video quality as compared with the conventional pixel domain approach(CPDT+FDVS). It is due to the fact that direct addition of DCT coefficients can reduce the re-encoding error significantly. However, this technique only can applied to Non-MC macroblocks. Conversely, our proposed algorithm can transcodes dominant regions of MC macroblocks in DCT domain (MCDCT) to avoid quality degradation. More importantly, the computational saving will be up to 60% as compared with the DCT-based video transcoder when the percentage of motion compensated macroblcok is increased.

4. CONCLUSION

In this paper, we have proposed a new architecture for a low-complexity and high quality frame-skipping transcoder to resolve the problem of MC macroblcoks transcoding. Its low complexity is achieved by: 1) re-using the DCT coefficients for macroblocks coded with motion compensation to deactivate most of the complex modules of the transcoder, 2) proposing an adaptive MC macroblock boundary transcoding and error compensation using selective inverse 1D-DCT, 1D-DCT, quantization and inverse quantization to process motion-compensated macroblocks. Furthermore, we have also shown that a direct addition of the DCT coefficients on macroblocks with motion compensation in dominant regions can reduce significantly the re-encoding errors due to transcoding. The overall structure of the proposed architecture produces a better picture quality than the conventional frame-skipping transcoder and DCT-based transcoder using the same reduced bitrates. Besides, the performance of the front encoder with half pixel accuracy is also evaluated. Since the percentage of MC-macroblocks becomes significant, our proposed transcoder outperform the DCT transcoder in terms of the video quality and computational complexity. It is due to the fact that our proposed transcoder decomposes MC macroblocks in the dominant region and boundary region. Therefore, low computational complexity can be achieved since only the boundary region is required to transcode in pixel-domain and perform error compensation. In other words, we can perform transcoding process in most regions of the DCT domain and avoid quality degradation. Experimental results also confirm that our proposed MCDCT transcoder gives outstanding performance for transcoding MC macroblocks which are the basis for hybrid video coding.

Table 5. Average PSNR a of various dynamic transcoders as compared with CPDT+FDVS using H.263 TMN8 [8] as a front encoder with half pixel precision.

Input bitrate	64kbps with 30 frames/s		128kbps with 30 frames/s
Sequence	Method	Average PSNR	Average PSNR
Salesman	CPDT+FDVS	33.41	36.91
	DA + FDVS + EC	34.72	38.23
	DA + FDVS + MCDCT + EC	35.74	39.33
Foreman	CPDT+FDVS	31.05	34.63
	DA + FDVS + EC	32.51	36.01
	DA + FDVS + MCDCT + EC	33.95	37.71
Car phone	CPDT+FDVS	32.74	35.38
	DA + FDVS + EC	33.67	36.02
	DA + FDVS + MCDCT + EC	35.16	37.80

5. ACKNOWLEDGMENTS

This work is supported by the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, Hong Kong Polytechnic University. K.T. Fung acknowledges the research studentships provided by the same University.

6. REFERENCES

[1] M.S.M. Lei, T.C. Chen, and M.T. Sun, "Video bridging based on H.261 standard," IEEE Trans. Circuit Theory, vol. 4, pp. 425-437, Aug. 1994.

[2] Neri Merhav and Vasudev Bhaskaran. "Fast Algorithms for DCT-Domain Image DownSampling and for Inverse Motion Compensation," IEEE Trans. On Circuit and systems for Video Tech., vol.7, no.3, June 1997.

[3] G. Keeman, R. Hellinghuizen, F. Hoeksema and G. Heideman, "Transcoding of MPEG-2 bitstreams," Signal Processing: Image Communication, vol. 8, pp. 481-500, Sept. 1996.

[4] Jeongnam Youn, Ming-Ting Sun and Chia-Wen Lin, "Motion vector refinement for high-performance transcoding," IEEE Trans. Multimedia, vol. 1, pp. 30-40, March 1999.

[5] K. T. Fung, Y. L. Chan and W. C. Siu, "New architecture for dynamic frame-skipping transcoder," IEEE Trans. Image Processing, vol.11, pp. 886-900, August 2002.

[6] Y.-Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding – Part II: Inter-frame prediction," IEEE Trans. Broadcast., vol. 37, pp. 102-105, Sept. 1991.

[7] J. Chalidabhongse and C.-C. Jay Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," IEEE Trans. Circuits Syst. Video Technol., vol. 7, pp. 477-488, June 1997.

[8] ITU-T/SG15, "Video codec test model, TMN8," June 97.