New DCT-Domain Transcoding using Split and Merge Technique

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

For the conventional downsampling video transcoder, the video server would be to first decompress the video, perform the downsampling operation in the pixel domain, and then recompress it. This is computational intensive. However, it is difficult to perform video downsampling in the DCT-domain since the prediction errors of each frame are computed from its immediate past higher resolution frames. Besides, the motion vector need to resample due to the size of the video is changed. Due to the mismatch of the resampled motion vector with the incoming DCT coefficients, the video transcoder need to recalculate the new DCT coefficient with lower resolution in pixel domain; this can create undesirable complexity as well as introduce re-encoding error. In this paper, we propose a new architecture to obtain the new DCT coefficients and the new motion vector by reuse the incoming motion vector and DCT coefficients. Since our proposed transcoder is mainly performing in DCT domain, low computational complexity can be achieved as well as the re-encoding can be reduced. Experimental result show that our proposed video downsampling transcoder can lead to significant computational savings as well as achieve high video quality compared to the conventional approach.

1. Introduction

Transcoding techniques are becoming an important role for a video server to provide quality support services to heterogeneous clients or transmission channels[1-7]. In this scenario that the video server should have capability of performing transcoding using different transcoding approaches to converting a previously compressed video bitstream into a lower bitrate bitstream without modifying its original structure according to the client devices (e.g. mobile communication) in terms of its computational complexity and bandwidth constraint. Several transcoding techniques are proposed recently such as requantization[1], frame rate reduction[2] and video downsampling[3-5] to reduce the bitrate of the incoming bitstream. In this paper, we will focus on the technique for video downsampling. One straightforward approach for implementing transcoding is to cascade a decoder and an encoder[1], commonly known as pixel-domain transcoding. To downscale the encoded video produced by the current video compression standards such as MPEG, H.261 or H.263 which employ motion compensated prediction to exploit the temporal redundancy to achieve a lower bitrate. The conventional approach needs to decompress the video and performs downsampling of the video in pixel domain. Then new motion vectors and DCT coefficients for this downscaled video are needed to recompress inside a transcoder. This involves a high computational complexity, memory, and long delay on a video server to generate the downscaled video. One of the simplest approach to reduce the computational complexity may be take the average of the four motion vectors associated with the four macroblocks and halve it so that the resulting motion vector can be associated with the 16x16 macroblock of the downscaled by two video. The motion vectors obtained in this manner are not optimal[3]. As a consequence, some information reusing approaches [3-5] such as adaptive motion vector resampling [3] for downscaled videos were used to provide a computational efficient solution to recompose the new motion vector. However, these approaches only deal with the problem of motion re-estimation during the transcoding process, the DCT coefficients with lower resolution are required to be recomputed. Due to the mismatch of the resultant motion vector with the incoming DCT coefficients, the video transcoder has to recalculate the new DCT coefficients with lower resolution in pixel domain; this can create undesirable complexity as well as introduce re-encoding error. The effect of re-encoding errors is depicted in Figure 1 where the "Table Tennis" sequence was transcoded at quarter of the incoming frame size. This figure shows that re-encoding errors lead to a significant degradation in the picture quality.

2. Low Complexity and High quality Video Downscaling in DCT Domain Transcoding

In this section, we present a new DCT-based video downsampling transcoding architecture. The new architecture has focus on the following areas:

2.1 Transcoding the overlapping region of MC macroblocks in DCT domain using the minimum distance motion vector with exchange video resolution (MVMD)

2.2 Adaptive transcoding of the non-overlapping region MC macroblocks boundary
2.3 Reconstruction of the new prediction errors and adaptive re-encoding error control architecture

The architecture of the proposed transcoder is shown in Figure 2. 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. Each macroblock is then manipulated independently. Switches, $S_{IP}$, are employed to pass the reconstructed quantized DCT coefficients to the DCT-domain downsampling operator, for the transformed and quantized residual signal depending on the coding mode originally used in the front encoder for the current macroblock being processed. For non-MC macroblocks or well-align cases, the incoming prediction errors in the UCT-domain is calculated by the DCT coefficients in the boundary region. The frame buffer can be reused. In other words, full inverse DCT, forward DCT, motion vectors from the incoming bitstream are $B_0, B_1, B_2$, and $B_3$ having one motion vector. And the other blocks have other three different motion vectors. Since our proposed new motion vector at the boundary is the minimum distance motion vector and the shifted operator to compute the DCT coefficients in overlapping region in DCT domain to achieve low computational complexity and avoid the re-encoding error.

Figure 3 shows the scenario that the macroblock is pointing to the overlap region in the previous frame. The prediction error obtained from the incoming bitstream are $B_0, B_1, B_2$, and $B_3$ as shown in Figure 4a, where $B_0, B_1, B_2$, and $B_3$ have one motion vector. And the other blocks have other three different motion vectors. Since our proposed new motion vector before downscaling is different from the incoming motion vectors as shown in Figure 3, the prediction error $B_0', B_1', B_2'$ and $B_3'$ reference the previous frame as shown in Figure 3 is needed to obtain in order to employ the downscaling of DCT coefficients in DCT domain. Our proposed resampled motion vector can be obtained as described in the following equations:

$$\frac{\partial}{\partial x} \text{Cost}_f(x) = -2(x-x_1) - 2(x-x_2) - 2(x-x_3) - 2(x-x_4)$$

$$\frac{\partial}{\partial y} \text{Cost}_f(x) = -2(y-y_1) - 2(y-y_2) - 2(y-y_3) - 2(y-y_4)$$

where $x$ and $y$ are resampled motion vector in horizontal and vertical direction respectively and $(x_1, y_1), ..., (x_4, y_4)$ are the motion vectors from the incoming bitstream as shown in Figure 3. Note that if all the incoming motion vectors are the same, i.e., $x_1 = x_2 = x_3 = x_4$ and $y_1 = y_2 = y_3 = y_4$, the new motion vector will be $(x_1, y_1)$. Otherwise, we need to minimize the cost function in both vertical and horizontal direction as shown below:

$$x = \frac{y_1 + y_2 + y_3 + y_4}{4}$$

Then the new prediction error $B_0', B_1', B_2'$ and $B_3'$ employs this resampled motion vector $(x, y)$ with minimum distance ($dx-x_1$ and $dy-y_1$) from the original motion vector $(x, y)$ to avoid the re-encoding error in the overlapping region if $B_0', B_1', B_2'$ and $B_3'$ can be obtained in DCT domain directly from $B_0, B_1, B_2, B_3, B_4, B_5, B_6, B_7$ and $B_8$ as shown in Figure 4a without performing requantization process during transcoding.

Since the shifted version of the original motion vector is used, the corresponding prediction error in the overlapping region can be
obtained without performing full re-encoding process. If the other three motion vectors have the same direction (i.e. well-aligned), the whole new prediction (B₂', B₃', and B₄') can be obtained in the DCT domain. In other words, no re-encoding error will be obtained without performing full re-encoding process. If the other three motion vectors have the same direction (i.e. well-aligned), the whole new prediction (Bo', BI', B2' and B3') is constructed using the pre-computed DCT values of the shifting matrices. Then the DCT values of the shifting matrices can be extracted from the incoming bitstream, so the re-encoding and high computational complexity can be avoided. If the macroblocks MB₁(1,1), MB₁(2,1), MB₁(3,1), and MB₁(4,1) have the same motion vector, direct downsampling of DCT coefficients can be applied in the DCT domain. Otherwise, decomposition of the overlapping region and boundary region is needed as shown in Figure 3.

Consider Pₖ be the targeted block of interest in pixel domain as shown in Figure 4a, P₀, P₁, P₂ and P₃ are the four neighboring blocks in pixel domain from which Pₖ is derived, and the new resampled motion vector obtained using minimum distance derivation approach is (x,y) with a minimum distance (dx=x₀-xₖ and dy=y₀-yₖ) from the original one. The shaded regions in P₀, P₁, P₂ and P₃ are moved by dx and dy. Then Pₖ can be represented in the following equation.

\[ P'_k = \sum_{i=0}^{1} S_{i} P S_{n} \]  

where Sₙ are matrices tabulated in Table 1.

![Figure 4a](image)

**Figure 4a** Incoming DCT coefficients of four macroblocks. 4b) The overlapping region and the boundary region of MB₁,

<table>
<thead>
<tr>
<th>Sub-block</th>
<th>P₀</th>
<th>P₁</th>
<th>P₂</th>
<th>P₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₀</td>
<td>0 Iₘₚ</td>
<td>0 Iₘₚ</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S₁</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Each I is an identity matrix of size 8x8 or 16x16 or 32x32 or 64x64. The post-multiplication shifts the sub-block of interest horizontally while post-multiplication shifts the sub-block vertically. Four possible locations of the sub-block of interest: upper-left, upper-right, lower-left and lower-right are shown in Table 1.

Define the 2D-DCT of an 8x8 block A as

\[ DCT(A) = A T A'^T \]  

where T is the 8x8 DCT matrix with entries \( k(i,j) \) given by

\[ k(i,j) = \frac{1}{2\sqrt{2}} \cos \left( \frac{2\pi i j}{16} \right) \]  

where \( i \) represents the row index and \( j \) represents column index and

\[ k(i) = \begin{cases} \frac{1}{\sqrt{2}}, & i = 0 \\ 1, & \text{otherwise} \end{cases} \]  

Using the properties of DCT,

\[ DCT(CD) = DCT(C) DCT(D) \]  

The major idea to obtain the DCT coefficients in overlapping region is to represent it as an addition of horizontally and vertically displaced anchor blocks. Then the DCT values of the overlapping region and boundary region are needed.

From eq(6), we have,

\[ DCT(P'_k) = \sum_{i=0}^{1} DCT(S_i) DCT(P_i) DCT(S_n) \]  

Note that the DCT of the S₀ and S₁ can be pre-computed and B₀ can be extracted from the incoming bitstream, so the re-encoding and high computational complexity can be avoided. If the macroblocks MB₁(1,1), MB₁(2,1), MB₁(3,1), and MB₁(4,1) have the same motion vector, direct downsampling of DCT coefficients can be applied in the DCT domain. Otherwise, decomposition of the overlapping region and boundary region is needed as shown in Figure 3.

Using the properties of DCT,

\[ DCT(A+B) = DCT(A) + DCT(B) \]  

where A and B represent the pixels in overlapping region and boundary region with size 8x8 respectively. We can split the block B₀', B₁' and B₂' in two regions: overlapping region and boundary region as shown in Figure 3.

Then the overlapping region of B₀ can be obtained by considering the boundary B₀ and B₁ with zero DCT coefficients value as shown in Figure 3. Similarly, the overlapping region of B₁' and B₂' can be obtained by considering the boundary B₀ and B₁ with zero DCT coefficients value and the boundary B₀, B₁ and B₂ with zero DCT value as shown in Figure 3. Then the DCT value of the overlapping region A can be obtained. Note that the DCT value of B₀', B₁' and B₂' is not completed since the DCT in boundary region B is not considered as described in equation (13). In the following, the DCT value of boundary region B in 8x8 is need to calculate it separately by using 1-D inverse DCT, motion compensation, forward ID-DCT and requantization as shown in Figure 3. In this process, re-encoding error cannot be avoided due to requantization.

### 2.2 Adaptive transcoding of the non-overlapping region MC macroblocks boundary

For the transcoding of MC boundary, the boundary region is extracted and selective quantization and selective ID-DCT of the quantized DCT coefficients of MB₁(1,1), MB₁(2,1), MB₁(3,1) and MB₁(4,1) are only needed to perform in the boundary as shown in Figure 3. Note that each macroblock is composed of four 8x8 blocks in video coding standards, and the DCT and quantization operations are performed on units of 8x8 blocks. When processing MB₁(1,1), MB₁(2,1), MB₁(3,1) and MB₁(4,1), only their corresponding 8x8 blocks, which have pixels overlapping with MB₁ boundary, are subject to the selective inverse 1D-DCT computation. Hence, the partially inverse 1D-DCT is only performed in the boundary, so the motion vector \( \text{mv}_{1,1} \) is needed as an input to the adaptive 1D-DCT module to control which rows or columns are needed to perform the 1D-DCT as shown in Figure 2. In most cases, this approach significantly reduces the required number of columns or rows DCT compare with the 2D-DCT approach.

Therefore, the new DCT coefficients of the boundary region can be obtained using partially transformed by putting the zero
values in the overlapping region and perform adaptive quantization to achieve low computational complexity.

2.3 Reconstruction of the new prediction errors and adaptive re-encoding error control architecture

After DCT coefficients of region B are obtained, the \( B_2', B_3', \) and \( B_4' \) can be reconstructed by adding DCT(A) and DCT(B) together using equation (13). In Figure 2, the newly quantized DCT coefficient of a MC macroblock can then be performed downscaling of DCT coefficients in DCT domain as described in Figure 3. For well-aligned case, the downscaling of the incoming DCT coefficients can be performed in DCT domain directly. Conversely, requantization is performed for the formation of new DCT coefficients in the macroblock boundary in not well-aligned case, which will introduce additional re-encoding errors.

Note that the re-encoding error only introduced in the boundary region as shown in Figure 3. However, these errors will degrade the quality of the reconstructed frame. Since each P-frame is used as a reference frame for the following P-frame, quality degradation propagates to later frames in a cumulative manner. Thus, a feedback loop is suggested as shown in Figure 2 to compensate for the re-encoding errors introduced in the boundary region. The adaptive forward and inverse DCT and adaptive quantization pairs in the feedback loop are mainly responsible for minimizing re-encoding errors. For these MC macroblocks, the requantized DCT coefficients are adaptive inversely quantized and adaptive inverse DCT is then performed. The re-encoding error introduced in the boundary region can be obtained by subtracts the original signal and the recovered signal after quantization. This re-encoding error is then stored in \( P \) and feedback to the latter frames to avoid the accumulation of re-encoding error.

After the reconstruction of the new prediction errors, the DCT domain downsampling will be performed and the resampled motion vector will be downsampled. After downscaling process, variable length encoding is applied. Then the output data is stored into output buffer for transmission.

3. Experimental Results

Results of the simulations are used to compare the performance of a reference transcoding (CPDT) by employing align-to-average weighting (AAW), align-to-best weighting (ABW) or adaptive motion vector resampling (AMVR) to resample a downsampled motion vector from the incoming motion vectors of the four macroblocks. In the front encoder, the first frame was encoded as intraframe (I-frame), and the remaining frames were encoded as interframes (P-frames). Picture-coding modes were preserved during transcoding. We show that our proposed DCT-based transcoders outperform CPDT+AAW, CPDT+ABW and CPDT-AMVR in all cases as shown in Table 2. The results are more significant for the low motion activity sequences because our proposed DCT-based transcoder should not introduce any re-encoding errors in the overlapping region since the transcoding is performed in DCT domain. Therefore, significant improvement in MC regions can be achieved about 1.25-1.4 times as compared with the conventional video downsampling transcoder. Also, our proposed transcoders have a speed-up of about 1.5-2.3 times faster than that of the conventional transcorder. This is because we can achieve significant computational savings while maintaining good

### Table 2: Average PSNR of the proposed transcoding, where the frame rate of the incoming stream was 30 frames.

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Input bitrate</th>
<th>Average PSNR difference as compared with CPDT+AAW for MC macroblock transcoding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPDT+AAW</td>
</tr>
<tr>
<td>Selected</td>
<td>512K</td>
<td>0.56</td>
</tr>
<tr>
<td>Miss_American</td>
<td>1536K</td>
<td>0.99</td>
</tr>
<tr>
<td>Hall</td>
<td>512K</td>
<td>0.11</td>
</tr>
<tr>
<td>Tennis</td>
<td>3M</td>
<td>0.12</td>
</tr>
<tr>
<td>Football</td>
<td>3M</td>
<td>0.27</td>
</tr>
<tr>
<td>Flower</td>
<td>3M</td>
<td>0.08</td>
</tr>
</tbody>
</table>

4. Conclusion

In this paper, we have proposed a new architecture for a low-complexity and high-quality video downsampling transcoder to solve the problem of MC macroblocks transcoding. Its low complexity is achieved by: 1) Re-use the DCT coefficients for macroblocks coded with motion compensation to deactivate most of the complex modules of the transcoder and 2) an adaptive MC macroblock boundary transcoding and error compensation using selective inverse DCT-DCT quantization and inverse quantization for motion-compensated macroblocks. Furthermore, we have also shown that transcoding the overlapping region of MC macroblocks in DCT domain using the minimum distance motion vector with unchance video resolution, an adaptive transcoding of the non-overlapping region MC macroblocks boundary and reconstruction of the new prediction errors and adaptive re-encoding error control architecture can reduce significantly the re-encoding errors due to transcoding. The overall performance of the proposed architecture produces a better picture quality than the conventional video downsampling transcoder at the same reduced bitrates.

5. Acknowledgements

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

6. References


