Low-Complexity and High Quality Frame-Skipping Transcoder

Kai-Tat Fung, Yui-Lam Chan 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

Transcoding is a process of converting a previously compressed video bitstream into a lower bit-rate bitstream. When some incoming frames are dropped for the frame-rate conversion in transcoding, the newly quantized DCT coefficients of prediction error need to be re-computed. In this paper, we propose a new architecture for low-complexity frame-rate reduction. The proposed algorithm is mainly performed on the discrete cosine transform (DCT) domain to achieve low complexity as well as to reduce the re-encoding error. Experimental results show that, as compared to the conventional transcoder, the new frame-skipping transcoder is more robust, produces smaller requantization errors, and has simple computational complexity.

1. Introduction

To transport video over low bandwidth channels, a high transcoding ratio is required. However, the high transcoding ratio may result in an unacceptable picture quality when the incoming bitstream is transcoded at the full frame-rate[1-2]. Frame skipping[1-4] is often used as an efficient scheme to allocate more bits to the remaining frame, so that an acceptable quality for each frame can be maintained.

One straightforward approach for implementing a transcoding is to cascade a decoder and an encoder, commonly known as pixel-domain transcoding. The incoming bitstream is decoded in the pixel domain, and the decoded video frame is re-encoded at the desired frame rate. This involves high processing complexity, memory, and delay. As a consequence, some information reusing approaches[1-2] have been proposed, in which some information such as motion vectors extracted from the incoming bitstream after the decoding can be used to significantly reduce the complexity of the transcoding.

In recent years, the DCT-domain transcoding was introduced[5-6], under which the incoming bitstream is partially decoded to form the DCT coefficients and downscaled by the requantization of the DCT coefficients. Since the DCT-domain transcoding is carried out in the coded domain where complete decoding and re-encoding are not required, the processing complexity is significantly reduced. However, the frame-rate conversion has not been fully considered. When the frame rate changes, the incoming quantized DCT coefficients of residual signal are no longer valid because they refer to the frames which have been dropped. Thus, it is difficult to perform frame-skipping in the DCT-domain since the prediction error of each frame is computed from its immediate past frames. In this paper, we provide a computationally efficient solution to perform the frame-skipping in a transcoder, mainly in the DCT-domain, to avoid the complexity arising from pixel-domain transcoding.

2. Pixel-Domain Transcoder

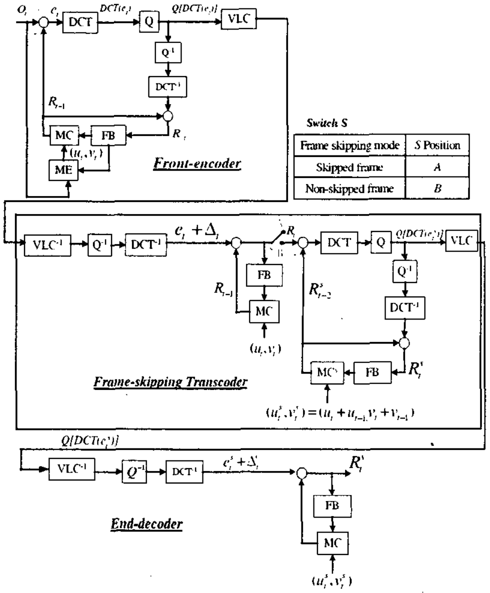

Figure 1. Frame-skipping transcoder in pixel-domain.

Figure 1 shows the structure of the conventional frame-skipping transcoder in pixel-domain. Switch *S* is used to control the desired frame rate of the transcoder. Assume that frame *t-1*, R_{t-1}, is skipped. However, R_{t-1} is required to act as the reference frame for the reconstruction of frame t, R_t, such that

$$R_t(i,j)=R_{t-1}(i+u_t, j+v_t)+e_t(i,j)+\Delta_t(i,j) \tag{1}$$

where (u_t, v_t) is the motion vector, $\Delta_t(i,j)$ represents the reconstruction error of the current frame in the front-encoder due to the quantization, and $e_t(i,j)$ is the residual signal between the current frame and the motion-compensated frame,

$$e_t(i,j)=O_t(i,j)-R_{t-1}(i+u_t, j+v_t) \tag{2}$$

Substituting (2) into (1), we obtain the expression for R_t,

$$R_t(i,j)=O_t(i,j)+\Delta_t(i,j) \tag{3}$$

In the transcoder, an optimized motion vector for the outgoing bitstream can be obtained by a new motion estimation. It is not desirable because of its high computational complexity. Reuse of the incoming motion vectors has been widely accepted because it is

0-7803-6685-9/01/$10.00©2001 IEEE

considered to be almost as good as performing a new full-scale motion estimation[1-2], thus,

$$(u_t^s, v_t^s) = (u_t + u_{t-1}, v_t + v_{t-1}) \tag{4}$$

Hence, the reconstructed pixel in the current frame after the end-decoder, $R_t^s(i,j)$, is,

$$R_t^s(i,j) = R_{t-2}^s(i+u_t^s, j+v_t^s) + e_t^s(i,j) + \Delta_t^s(i,j) \tag{5}$$

where $R_{t-2}^s(i,j)$ denotes a reconstructed pixel in the previous non-skipped reference frame, $\Delta_t^s(i,j)$ represents the quantization error of the current frame and the non-skipped reference frame due to the re-encoding in transcoder, and $e_t^s(i,j) = R_t(i,j) - R_{t-2}^s(i+u_t^s, j+v_t^s)$. The superscript "*s*" is used to denote the symbol after performing the frame-skipping transcoder. Then,

$$R_t^s(i,j) = O_t(i,j) + \Delta_t(i,j) + \Delta_t^s(i,j) \tag{6}$$

This equation implies that the reconstructed quality of the non-skipped frame deviates from the input sequence to the transcoder, $R_t(i,j)$. An additional error, $\Delta_t^s(i,j)$, is introduced. Re-encoding of the current frame involves a re-computation of the residual signal between the current frame and the non-skipped reference frame. Note that frame *t-2* acts as the reference instead of frame *t-1*, since frame *t-1* does not exist after frame skipping. The newly quantized DCT-domain data are then re-computed by means of the DCT and quantization process. This re-encoding procedure can lead to an additional error $\Delta_t^s(i,j)$. The effect of the re-encoding error is depicted in Figure 2 where the "Salesman" sequence was encoded. In the figure, the peak signal-to-noise ratio (PSNR) of the frame-skipping pictures is plotted to compare with that of the same pictures directly using a decoder without a transcoder. This figure shows that the re-encoding error leads to a drop in picture quality of about 3.5dB on average, which is a significant degradation.

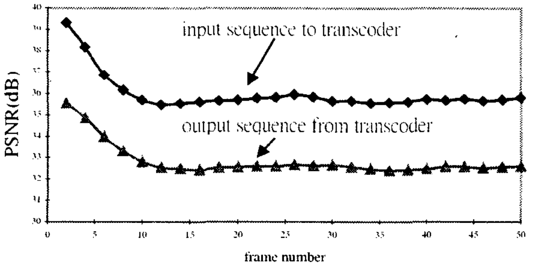

Figure 2. Quality degradation of conventional frame-skipping transcoder for "Salesman" sequence.

3. Proposed Frame-Skipping Transcoder

Besides the quality issue mentioned above, the pixel-domain transcoder also has a high processing complexity. This is due to the fact that the skipped frame must be decompressed completely, and should act as the reference frame to the non-skipped frame for reconstruction. In this section, we present a new architecture for frame-rate reduction in order to achieve an improved picture quality and a reduced complexity.

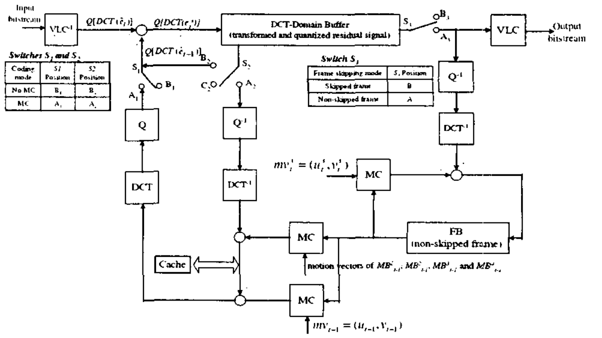

Figure 3. The proposed frame-skipping transcoder.

The architecture of the proposed transcoder is shown in Figure 3. The input bitstream is first parsed with a variable-length decoder to extract the coding mode, motion vector and quantized DCT coefficients for each macroblock. Each macroblock is then manipulated independently. The two switches *S1* and *S2* are employed to update the DCT-domain buffer for the transformed and quantized residual signal depending on the coding mode originally used at the front encoder for the current macroblock being processed. When the macroblock is not motion compensated, the previous residual signal in the DCT-domain is directly fed back from the DCT-domain buffer to the summer, and the sum of the input residual signal and the previous residual signal in the DCT-domain is updated in the buffer. Note that all operations are performed in the DCT-domain, thus the complexity of the frame-skipping transcoder is reduced. Also, the quality degradation of the transcoder introduced by $\Delta_t^s(i,j)$ is avoided. When motion compensation is used, motion compensation, DCT, inverse DCT, quantization and inverse quantization modules are activated to update the DCT-domain buffer. Note that switch *S3* is used to control the frame rate and refresh the frame buffer for non-skipped frames. Another advantage of the proposed architecture is that when multiple frames are dropped, it can be processed in the forward order, thus eliminating the multiple DCT-domain buffers that are needed to store the incoming quantized DCT coefficients of all dropped frames. Thus, only one DCT-domain buffer is needed for all the dropped frames.

3.1 Direct summation of DCT coefficients for macroblock without motion compensation

In Figure 4, a situation in which one frame is dropped is illustrated. We assume that MB_t represents the current macroblock and MB_{t-1} represents the best matching macroblock to MB_t. Since MB_t is coded without motion compensation, the spatial position of MB_{t-1} is the same as that of MB_t, and MB_{t-2} represents the best matching macroblock to MB_{t-1}. Since R_{t-1} is dropped, for MB_t, we need to compute a motion vector, (u_t^s, v_t^s), and the prediction error in quantized DCT-domain, $Q[DCT(e_t^s)]$, by using R_{t-2} as a reference. Since the motion vector in MB_t is zero, it simply re-computes such a motion vector as, $(u_t^s, v_t^s) = (u_t, v_t)$. Since re-encoding can lead to an additional error, it could be avoided if $Q[DCT(e_t^s)]$ can be computed in the DCT-domain. For the sake of simplicity, we define the incoming residual signal with the quantization error to the transcoder as $\hat{e}_t = e_t + \Delta_t$. From Figure 4, the pixels of MB_t can be reconstructed by performing inverse quantization and inverse DCT of $Q[DCT(\hat{e}_t)]$ and summing this residual signal to pixels in MB_{t-1} which can be similarly reconstructed by performing inverse quantization and inverse DCT of $Q[DCT(\hat{e}_{t-1})]$ and summing this residual signal to pixels in the corresponding MB_{t-2}. However, by considering the linearity of inverse DCT and inverse quantization, we obtain,

$$Q[DCT(e_t^S)] = Q[DCT(\hat{e}_t)] + Q[DCT(\hat{e}_{t-1})] \quad (7)$$

Equation (7) implies that the newly quantized DCT coefficient $Q[DCT(e_t^s)]$ can be computed in the DCT-domain by summing directly the quantized DCT coefficients between the data in the DCT-domain buffer and the incoming quantized DCT coefficients, whilst the updated DCT coefficients are stored in the DCT-domain buffer, as depicted in Figure 3, when switches S_1 and S_2 are connected to B_1 and B_2 respectively. Since it is not necessary to perform motion compensation, DCT, quantization, inverse DCT and inverse quantization, the complexity is reduced. Furthermore, since re-quantization is not necessary for this type of macroblock, the quality degradation of the transcoder introduced by $\Delta_t^s(i,j)$ is also avoided. Figure 5 shows the distribution of the coding mode for the typical "salesman" sequence; it is clear that over 95% of the macroblocks are coded without motion compensation. By using a direct summation of DCT coefficients for non-moving macroblocks, the computational complexity involved in processing these macroblocks can be reduced significantly and the additional re-encoding error can be avoided.

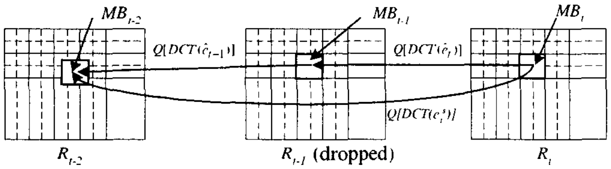

Figure 4. Residual signal re-computation of frame-skipping for macroblocks without motion compensation.

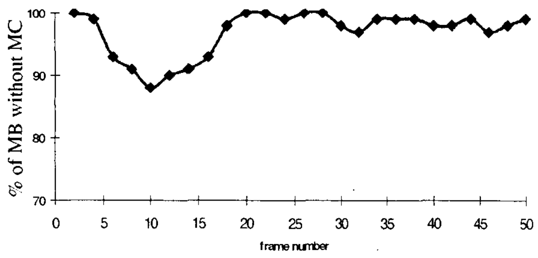

Figure 5. Coding mode without motion compensation.

3.2 DCT-domain buffer updating for motion-compensated macroblock

For motion-compensated macroblocks, direct summation cannot be employed since MB_{t-1} is not on a macroblock boundary, as depicted in Figure 6. In other words, $Q[DCT(\hat{e}_{t-1})]$ is not available from the incoming bitstream. It is possible to use the motion vectors and quantized DCT coefficients of the four neighboring macroblocks with MB_{t-1}, MB_{t-1}^1, MB_{t-1}^2, MB_{t-1}^3 and MB_{t-1}^4, to come up with $Q[DCT(\hat{e}_{t-1})]$. First, inverse quantization and inverse DCT of the quantized DCT coefficients of MB_{t-1}^1, MB_{t-1}^2, MB_{t-1}^3 and MB_{t-1}^4 are performed to obtain their corresponding prediction errors in the pixel-domain. Figure 6 shows that the MB_{t-1} is composed of four regions. Thus, each segment of the reconstructed pixels in MB_{t-1} can be obtained by summing its prediction errors and its motion-compensated segment of the previous non-skipped frame stored in the frame buffer, as shown in the block diagram of Figure 3. After all pixels in MB_{t-1} have been 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} subtracted from the motion-compensated macroblock from the previous non-skipped frame stored in the frame buffer. Since MB_{t-1} is not on a macroblock boundary, we need to find a motion vector of MB_{t-1} in order to obtain the motion-compensated macroblock. The dominant vector selection approach [1-2], which selects one dominant motion vector from the four neighboring macroblocks, is employed. A dominant motion vector is defined as the motion vector carried by a dominant macroblock. The dominant macroblock is a macroblock that has the largest overlapped segment with the MB_{t-1}. Hence, $\hat{e}_{t-1}$ is transformed and quantized to form $Q[DCT(\hat{e}_{t-1})]$. Since quantization is performed in the formation of $Q[DCT(\hat{e}_{t-1})]$, some quantization error, $\Delta_{t-1}^s(i,j)$, is introduced. The newly quantized DCT coefficient of a motion-compensated macroblock can be computed by

$$Q[DCT(e_t^S)] = Q[DCT(\hat{e}_t)] + Q[DCT(\hat{e}_{t-1})] + \Delta_{t-1}^S(i,j) \quad (8)$$

In most cases, $\Delta_{t-1}^s(i,j)$ is smaller than $\Delta_t^s(i,j)$ since the distance to the reference frame is one frame less.

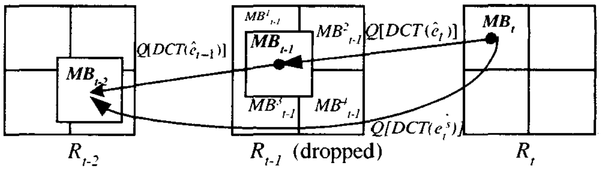

Figure 6. Residual signal re-computation of frame-skipping for motion-compensated macroblocks.

In order to reduce the implementation complexity of the motion-compensated macroblock, a cache subsystem is added to our proposed transcoder, as depicted in Figure 3. Since motion compensation of multiple macroblocks may require the same pixel data, a cache subsystem is implemented to reduce redundant inverse quantization, inverse DCT and motion compensation computations.

4. Simulation Results

The proposed frame-skipping transcoder was implemented according to the H.263. All test sequences of QCIF were encoded at high bitrate (64kbps and 128kbps) using a fixed quantization parameter. At the front encoder, the first frame was coded as intraframe (I-frame), and the remaining frames were encoded as interframes (P-frames). These picture-coding modes were preserved during the transcoding.

In this section, we evaluate the overall efficiency of the proposed transcoder. The PSNR performance of the proposed frame-skipping transcoder is shown in Figure 7. The original test sequence "Salesman" was encoded at 128kbps in the front encoder, and transcoded into 64kbps at half of the incoming frame-rate. As shown in the figure, the proposed transcoder outperforms the conventional pixel-domain transcoder. Also, Table 1 shows that it has a speed-up of 2-7 times faster than that of the conventional transcoder. This is because the probability of the macroblock coded without motion compensation happens more frequently in typical sequences. The direct summation of the DCT coefficients can be applied and we could achieve significant computational savings while maintaining good video quality. Also, the cache system in the transcoder can reduce the computational burden of re-encoding the motion-compensated macroblocks. All these advantages combined gives rise to significant computational saving. These demonstrate the effectiveness of the proposed frame-skipping transcoder. The simulation results of other test sequences are summarized in Table 2.

5. Conclusion

We have proposed a low-complexity and high quality frame-skipping transcoder. Its low complexity is achieved by: 1) a direct summation of the DCT coefficients for macroblocks coded without motion compensation to deactivate most complex modules of the transcoder, and 2) a cache subsystem for motion-compensated macroblocks to reduce redundant IDCT and inverse quantization. We have also shown that a direct summation of the DCT coefficients can eliminate the re-encoding error due to requantization. Furthermore, our proposed frame-skipping transcoder can be processed in the forward order when multiple frames are dropped. Thus, only one DCT-domain buffer is needed to store the updated DCT coefficients of all dropped frames. Overall, the proposed frame-skipping transcoder produces a better picture quality than the conventional frame-skipping transcoder at the same reduced bitrates.

6. Acknowledgments

This work is supported by the Centre for Multimedia Signal Processing, HKPolyU. K.T. Fung acknowledges the research studentships provided by the University and Dr. Y.L.Chan thanks the HKPolyU for the support he receives under its research fellowship scheme.

7. References

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

[2] Jeongnam Youn, Ming-Ting Sun and Chia-Wen Lin, "Motion estimation for high performance transcoding," IEEE Trans. on Consumer Electronics, vol. 44, pp. 649-658, August 1998.

[3] S.H.Kwok, W.C.Siu and A.G.Constantinides, "Adaptive temporal decimation algorithm with dynamic time window," IEEE Trans. on Circuits and Syst. for Video Tech., vol.8, pp. 104-111, Feb. 1998.

[4] S.H. Kwok, W.C.Siu and A.G.Constantinides, "A scaleable and adaptive temporal segmentation algorithm for video coding," Graphical Models and Image Proc., vol.59, pp. 128-138, May 1997.

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

[6] H. Sun, W. Kwok and J.W. Zdepski, "Architectures for MPEG compressed bitstream scaling," IEEE Trans on Circuits and Syst. for Video Tech., vol. 6, pp. 191-199, April 1996.

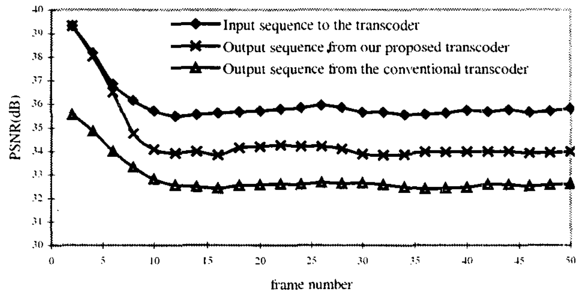

Figure 7. Performance using different transcoders with one-frame dropping of "Salesman" sequence.

Table 1. Speed-up ratio of the proposed transcoder.

Sequences	Input bitrate	Speed-up ratio
Miss American	128k	2.39
	64k	2.08
Foreman	128k	3.23
	64k	3.87
Salesman	128k	7.64
	64k	6.75

Table.2 Average PSNR comparison of various test sequences. The output bitrate is the half of the input bitrate.

Sequences	Input bitrate	Conventional transcoder	Our proposed transcoder
Miss American	128k	37.20	39.71
	64k	36.92	37.56
Foreman	128k	34.31	35.13
	64k	31.13	31.36
Salesman	128k	36.96	37.81
	64k	32.85	34.51