A VERY FAST VIDEO SPATIAL RESOLUTION REDUCTION TRANSCODER

Bo Shen, Sumit Roy
Hewlett-Packard Laboratories
{boshen, sumit}@hpl.hp.com

ABSTRACT

With the introduction of the next generation wireless network, mobile devices access increasingly more media-rich content. However, a key factor that prevents a mobile device from accessing multimedia content is that it does not have enough display real estate to render the content that is traditionally created for the desktop web client. Moreover, the wireless network typically has lower bandwidth compared to a wired network of the same time frame. Therefore, a transcoder is needed somewhere in the network to transform multimedia content to the appropriate form factor and bandwidth requirement. This paper introduces an extremely fast transcoder of such kind. Comparing to the previous methods, additional computation is saved while the quality of transcoding is maintained. This saving in computing resources maps into the benefit that more concurrent sessions can be supported on one transcoding device. This scalability is crucial for the wireless network to handle user requests that might be very intensive at times.

1. INTRODUCTION

The Internet brings heterogeneous devices together. For rich media transmission over the wireless Internet, content adaptation is a key issue. The original content may have been coded at higher resolution and higher bit rate, say 720x480 at 4 Mbps for DVD quality, or 320x240 at 1 Mbps for desktop clients connected to the Internet through T1 line. However, due to the characteristics of the mobile communication – low bandwidth channel and limited display real estate, a 100kbps video at a lower resolution is desired. Current 3G wireless communication targets at providing a 128~384Kpbs communication channel. Therefore, a transcoder is needed in the network to adapt the content to the appropriate size and bit rate.

A straightforward method to perform this transcoding is to decode the original stream, down sample the decoded frames to a smaller size and re-encode to a lower bit rate. Considering a typical CCIR601 MPEG-2 video, it takes the full power of a 400Mhz CPU to perform real-time decoding. Encoding is even more expensive, which makes the straightforward method non-practical. Furthermore, if the transcoding is provided as a network service between content provider and content consumer, one transcoding unit is expected to able to support as many concurrent sessions as possible. This scalability is crucial for the wireless network to handle user requests that might be very intensive at times. Therefore, it is extremely worthwhile to develop fast algorithms to reduce the CPU load for such kind of transcoding session.

In a previous work [1], we used a compressed-domain approach, where the motion information in the original video is reused. The approach significantly improved the performance since the costly motion estimation process is eliminated. However, all the frames in the original video need to be reconstructed, or otherwise, a frequency domain intra version of the inter frames needs to be constructed based on the algorithms proposed in [2]. However, due to its irregular data access pattern, this process still consumes large amount of CPU cycles.

This paper lays out a further optimized video transcoding algorithm. A macroblock-aware approach is proposed to take advantage of compressed domain processing techniques. In this approach, the transcoder performs transcoding at the macroblock level, therefore is able to take advantage of different combinations of macroblock types to avoid reconstruction of the original frame. However, we still need the original size reference frame for the decoding of future frames in the original video. The solution is that we *approximate* the original frame by up sampling the down-sampled version. As a result, only one-fourth of the macroblocks need to be reconstructed in a down-sample-by-two operation. Considering the fact that we can utilize an optimized DCT domain down-sampling methods [3] plus that the up-sampling operation is an easily optimizable operation either by hardware DSP design or MMX implementation, the computational saving is significant. For inter-frames, similar algorithms can be used if a sufficient number of the involved macroblocks are intra macroblocks. The approximation by up sampling is based on the rationale that, to generate a down-sampled inter-frame, the quality degradation is negligible when performing motion compensation based on down-sampled reference frames.

The paper is organized as follows. In Section 2, we describe the algorithm in detail and propose a transcoding system. Performance analysis and testing results are presented in Section 3. We conclude in Section 4.

2. ALGORITHM & SYSTEM

2.1. Macroblock-aware transcoding algorithm

To explain in more detail how the macroblock-aware transcoding algorithm works, this section goes through a down-sampling-by-two operation on MPEG [4] video. Figure 1 shows the process for each type of pictures as well

as different combinations of macroblocks in an MPEG video down-sampling application. The relative size of blocks and frames is also illustrated to reflect the change before and after the processes. For I-picture, a DCT domain down-sampling method [3] can be used to generate a down-sampled version of the I-picture. A pixel domain version has to be generated afterwards as a reference for the future P- or B-pictures in the group of picture (see the motion compensation arrow). Subsequently, an up sampling is performed to approximate the pixel domain version in the original size. The original size picture is needed for the reconstruction of the future P- or B-picture (see the reconstruction arrow). In this figure, the dotted-line boxes correspond to the frame buffer for the original size video, and the shaded boxes to the frame buffer for the down-sampled version.

For P- and B-picture, a mode-decision module decides whether the output macroblock is coded as inter or intra macroblock. If there are one intra block and three forward-predicted ones in the input macroblocks, the mode-decision module may decide to code the output block as a forward-predicted one. In general, assuming k macroblocks are involved in generating one output macroblock, the mode decision module decides that if at least m out of k macroblocks are intra macroblocks, the output macroblock will be coded as intra. Otherwise, the output macroblock is coded as inter. If the output macroblock is coded as intra, we further define a quantity n, where $m<n<k$. Depending on the actual number of input intra macroblocks, the selection of n decides what procedure the transcoder takes to generate the output intra macroblock. In summary, we have the following four scenarios that the macroblock-aware transcoder handles differently.

- Case 1: the output MB is an intra MB, and m to n input macroblocks are intra.
- Case 2: the output MB is an intra MB, and n to $(k-1)$ input macroblocks are intra.
- Case 3: the output MB is an intra MB, and all k input macroblocks are intra.
- Case 4: the output MB is an inter MB.

Using the down-sample-by-two operation as an example, four macroblocks are involved in generating one output macroblock, i.e. $k=4$. If we further select $(m, n)=(2, 3)$, we can use IIFF-I, IIIF-I, IIII-I and IFFF-F to map the aforementioned four scenarios. These symbols represent different compositions of the four original macroblocks. For example, if there are one intra block and three forward-predicted ones among the four input macroblocks, a mode-decision module may decide to code the output block as a forward-predicted one, hence the symbol IFFF-F. The other three symbols can be interpreted similarly. Note that each symbol does not literately represent one exact combination of input macroblocks, other possible combinations can be handled based on the understanding of these four scenarios.

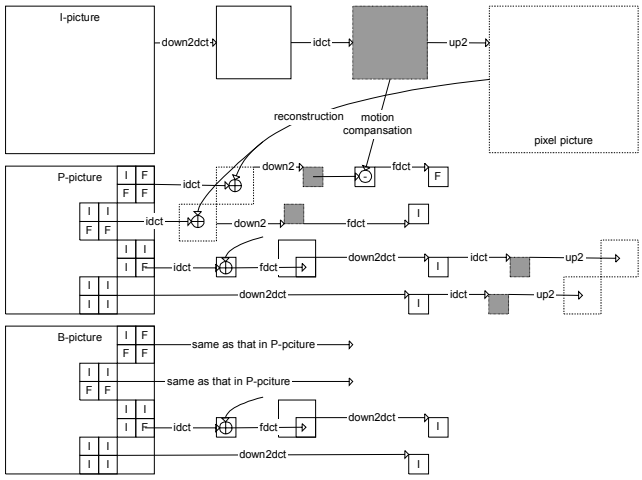

Figure 1 Data flow of the transcoder

To begin with, let us consider Case 4, the IFFF-F case. If the output macroblock is decided to be a predicted one, the original involved macroblocks are reconstructed regardless of whether the macroblocks are intra or non-intra. A spatial domain down sampling is performed on the reconstructed macroblock to generate a pixel domain output macroblock. Subsequently, the residual is generated based on the motion vector obtained from the original MPEG video. In this sequence of procedures, the reconstructed macroblocks along with subsequently down sampled version are kept as reference for the future P- or B-pictures. This scenario has no difference than a regular decoding and re-encoding scheme except that the motion information is reused instead of obtained through a costly motion estimation process.

If the output macroblock is decided to be an intra block, three different schemes are depicted in Figure 1. Case 3 represents the case when all four input macroblocks are intra (IIII-I). In this case, the DCT domain down sampling methods can be directly employed. Subsequently, the output macroblock is inverse discrete cosine transformed (IDCTed) to the pixel domain and the up-sampled version is also obtained for possible use as a reference frame for future pictures.

If only one of the four original is a non-intra macroblock (e.g., the IIIF-I scenario for Case 2), only the non-intra macroblock is reconstructed and a DCT version is generated through the forward discrete cosine transform (FDCT) process (this process can also be performed through a DCT domain motion compensation [2]). Then the four DCT macroblocks are used to construct one DCT macroblock using a DCT domain downscaling algorithm [3] (see the down2DCT arrow). Subsequently, the output macroblock is IDCTed to the pixel domain and the up-sampled version is also obtained for possible use as a reference frame for future pictures.

For Case 1, we use IIFF-I as an example. In this case, we reconstruct every involved macroblock regardless of

whether it is intra or not. Subsequently, a spatial domain down sampling is performed on the reconstructed macroblocks to generate a pixel domain output macroblock. The macroblock is then FDCTed to generate the output intra macroblock. In this case, we cannot take advantage of the DCT domain down-sampling method. The reason is that it is more costly to reconstruct the intra DCT version of the two input inter macroblocks.

For B-pictures, the four different scenarios described for P-pictures apply as well except that the reconstruction of some macroblocks may take reference from past and/or future pictures. In addition, as shown in the case of IIIF-I and IIII-I, we do not have to go through the processes of IDCT and up sampling for obtaining the pixel frame with original size. This is due to the fact that B-pictures are not used as reference frames.

The process for down sampling by a factor of three or four can be easily derived from the above. In down-sample-by-four case, k is equal to 16. If we select m as 9, we found that it is optimal to select n as 12. Similar scenarios from Case 1 to 4 can be identified and handled accordingly by the macroblock-aware transcoder. Note that since the down-sample-by-four operation brings in many more possible combinations of macroblocks, further optimization can be done for some cases depending on the locations of the involved intra macroblocks.

2.2. Transcoding system

Figure 2 illustrates the data flow of the transcoding process. From the input buffer, variable length code (VLC) decoder parses the input video stream. Motion vectors are passed to motion compensation module and MV Generator for generating new motion vectors for the downscaled version. The DCT data is sent to inverse quantizer and subsequently, a mode decision module decides whether the output macroblock is coded as inter or intra.

- If it is Case 2 and 3, the input DCT data is sent to the DCT domain down-sampling module to generate down sampled DCT data. For the macroblocks directly generated by DCT domain down sampling, IDCT is performed and the result is saved in Frame Buffer B. Furthermore, a up sampling is performed to generate a reconstructed version with original size that is saved in Frame Buffer A.
- Otherwise, the macroblock needs to be reconstructed. The DCT data is IDCTed and motion compensated if the macroblock is an inter macroblocks. The result is saved in Frame Buffer A. Therefore, the data in Frame Buffer A is the reconstructed macroblock with original size. It is then down sampled and put into Frame Buffer B.

The data in Frame Buffer A is used to reconstruct future frames in the original video. The data in Frame Buffer B is used to generate the new residual based on the new motion vector. The residual is subsequently FDCTed and sent along with the DCT data generated directly by DCT domain down-sampling module to the forward quantizer. The rate control module controls the step size of the quantizer in order to achieve specified output bit rate. Finally, VLC encoder generates the output binary bit stream.

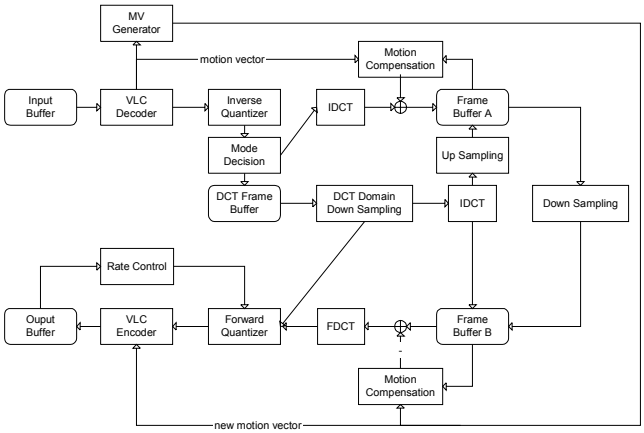

Figure 2 Processing flow of the transcoder

MV Generator is responsible to generate new motion vector based on the motion vectors from the original stream. The new motion vector could be a scaled version of the some weighted average of the original ones. A previous work [1] presents some analysis regarding the optimal way to generating the new motion vector.

This system represents a simplified version in which an open-loop approach is used. The open-loop approach may cause the error propagation problem on the receiver. However, the simplified system performs much faster due to the elimination of one inverse quantization process and one IDCT process per loop.

3. ANALYSES & TESTING RESULT

Primarily, the computational saving comes from the elimination of a number of IDCT processes. Here, one IDCT process represents the IDCT operations needed for the reconstruction of one macroblock. For 420-chrominance compression [4], six 8x8 IDCT operations are invoked for the reconstruction of one macroblock. For I-pictures, the number of IDCT operations required is $1/N^2$ times originally required, where N is the down-sampling factor. For example, for the down-sample-by-two operation, it originally needs four IDCT operations for one output macroblock, now only one IDCT is needed. Considering that one DCT domain down sampling process cost similar computation as one FDCT, three IDCTs are saved, which is about 75% saving. The same applies to the IIII-I case in P- and B-pictures. For the IIIF-F case in P- and B-pictures, two IDCTs are required instead of four. Considering an additional DCT domain down sampling process, one IDCT is saved, which is about 25% saving. Additional saving comes from the elimination of the reconstruction of pixel domain version of B-pictures for

IIIF-I and IIII-I cases. The saving is more significant for down-sample-by-three (N=3) and down-sample-by-four (N=4) cases.

The computational saving is video sequence dependent. For example, if a P-picture in the original stream contains mostly predicted macroblocks, the advantages brought by the smart handling of IIII-I or IIIF-I case are limited. However, since this algorithm is targeting at transcoding high bit-rate and high-resolution video to low bit-rate and low-resolution video, statistics show that many macroblocks are coded in intra mode in the original high bit-rate and high-resolution video. Figure 3 shows the percentage of different cases for each frame in the football test stream. The football test stream contains 150 frames of size 704x480. It is coded at 6 Mbps using IBBPBB… structure in groups of pictures that have 15 frames.

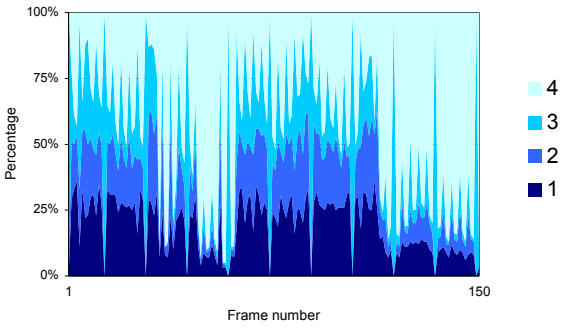

Figure 3 Stacked percentage of different cases

To test the performance of the macroblock-aware transcoder, we perform a down-sample-by-two operation on the test stream. The transcoder generates an output video that is coded at 1.5 Mbps with the size of 352x240. Two PSNR curves are shown in Figure 4. The old PSNR curve is obtained by using picture-level reconstruction in the transcoding; the new PSNR curve is obtained by using macroblock-aware transcoding proposed in this paper. Both PSNR are computed against the frames generated by decoding the original video and down sampling the result by two. A frame-by-frame quality comparison is given in Figure 5.

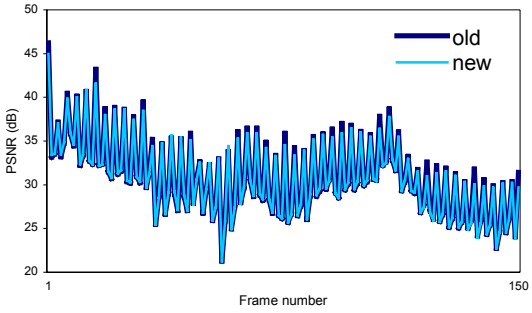

Figure 4 PSNR of each frame generated by the old and the new transcoder

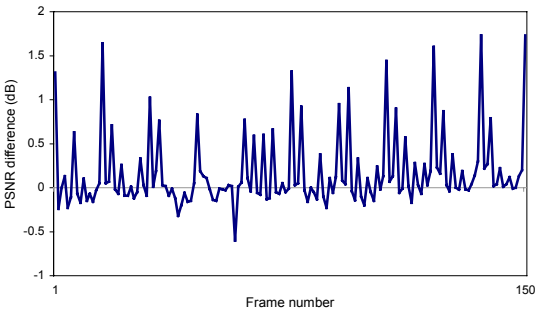

Figure 5 PSNR difference

The experiment indicates that the macroblock-aware transcoding renders similar quality output. Note that the PSNR drop for I-pictures is relatively larger. This is exclusively due to the approximation nature of the DCT domain down sampling algorithm that is used in the implementation.

4. CONCLUSION

Even by reusing the original motion information, the processing speed to spatial resolution reduction of compressed video is limited by the necessity of the reconstruction of original frames. This paper presented an optimized algorithm that avoids the reconstruction of original frames as much as possible. Moreover, the proposed algorithm takes advantage of fast DCT domain down sampling methods as much as possible without the reconstruction of intra DCT version of original frames. Therefore, additional computational saving is achieved with negligible loss of quality. The algorithm is implemented in an MPEG video transcoding system. The algorithm applies to other types of compressed video as well, as long as motion-compensated motion estimation and DCT-based frequency domain compression techniques are used in the compression.

5. REFERENCES

[1] B. Shen, I. Sethi and V. Bhaskaran, "Adaptive motion-vector resampling for compressed video downscaling," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 9, no. 6, pp. 926-936, Sept. 1999.

[2] N. Merhav and V. Bhaskaran, "A fast algorithm for DCT-domain inverse motion compensation," *Proc. ICASSP'96*, Atlanta, GA, pp. IV.2307-2310, May 1996.

[3] B. Natarajan and V. Bhaskaran, "A fast approximate algorithm for scaling down digital images in the DCT domain," *Proc. IEEE int. Conf. On Image Processing (ICIP)*, Washington DC. Oct. 1995.

[4] J. L. Mitchell, W. B. Pannebaker, C. E. Fogg and D. J. LeGall, MPEG Video Compression Standard, Chapman& Hall, 1995.