P-FRAME TRANSCODING IN VC-1 TO H.264 TRANSCODERS

Maria Pantoja[1], *Hari Kalva*[2], *and Jae-Beom Lee*[3]
[1]Dept. of Computer Science, Santa Clara University, Santa Clara, CA
[2]Dept. of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431
[3]Sarnoff Corporation, Princeton, NJ 08540

ABSTRACT

VC-1 is now one of the three video coding standards for high definition DVD that include MPEG-2 and H.264. The coded is expected to be used in consumer electronic devices such as DVD and camcorders. The H.264 format has begun to see strong acceptance and is used in mobile devices such as iPod and mobile phones. While multi-format DVD players are expected to support the three high definition video coding formats, H.264 is expected to have broader support in devices and video download services. The need to move data among devices with different capabilities creates a need for transcoding. In this paper we present a P-frame transcoder for VC-1 to H.264 transcoding. The transcoder exploits the variable size transform used in VC-1 to select the variable block size for motion compensation in H.264. The transcoder reduces the complexity substantially without significant loss in quality.

Index Terms— *H.264*, VC-1, transcoding, complexity reduction

1. INTRODUCTION

The VC-1 video compression standard developed by Microsoft has been standardized by SMPTE and adopted by the DVD forum for high definition DVDs. VC-1 is also expected to be deployed as a key engine in satellite TV, IP set-tops and high-definition DVD recorders. H.264 is an emerging standard that is being adopted for a broad range of video applications including digital TV and mobile video services. The coexistence of these different video coding standards creates a need for transcoding. This paper presents a VC-1 to H.264 video transcoder with reduced complexity.

The emergence of H.264 has resulted in increasing research in the area of transcoding to H.264 format. Transcoding tools and algorithms have been proposed to transcode video from H.263 [1], MPEG-4 [2], and MPEG-2 [3] to the H.264 format. Complexity reduction in transcoding is typically achieved by reusing the information from the decoding stage of the input video format. The transform domain approaches perform partial decoding of the incoming video and directly convert transform coefficients of the input video to the transform coefficients of the output video format. Work on converting MPEG-2 DCT coefficients to H.264 transform coefficients is reported in [4, 5]. In pixel domain transcoding, the input video is completely decoded and then encoded to a target format using a reduced complexity encoder that uses the information from the decoding stage. This approach has shown promising results and is reported in several papers [6-8].

While VC-1 and H.264 have significant differences, the two encoders still use hybrid coding techniques and have features that allow complexity reduction. While there has been recent work on MPEG-2 to H.264 transcoding, the published work on VC-1 to H.264 transcoding is minimal. There is very limited amount of published work on VC-1 [9]. Our prior work on VC-1 to H.264 transcoding that outlines the approaches for transcoding was reported in [11]. In this paper we present algorithms for P-frame transcoding and report the performance results. The proposed approach leverages the variable transform size used in VC-1 to determine the size of the blocks for motion estimation in H.264. With this approach, instead of evaluating all the possible block sizes for motion estimation, the transcoder evaluates a single block size determined based on the transform size used in VC-1.

The rest of the paper is organized as follows: Section 2 gives an overview of the VC-1 video compression and Section 3 presents the proposed MB mode selection in P frames. The experimental results and discussion is presented in Section 4 and conclusions are presented in Section 5.

2. COMPRESSION TOOLS FOR VC-1 PROGRESSIVE VIDEO

This section provides a brief overview of VC-1 with emphasis on the features that impact transcoding. Like most video coding standards, VC-1 is based on the principles of hybrid video coding: motion compensated transform coding. VC-1 has five picture types: I, P, B, BI and Skipped P. Similar to MPEG standards, an I frame has all MBs that are Intra coded, the P frame has MBs that are Intra or predicted from previous frames, and the B MBs are bi-directionally predicted. Unlike H.264, B frames cannot be used as reference frames. The BI frames are almost identical to I frames and are used in place of B frames. A BI frame is Intra-coded frame and is used instead of a B frame when the B frame coding is inefficient. If there is big continuous scene change, B frames cannot capture any similarity from two reference frames. In such cases, BI frame compression is a good choice. Since BI doesn't have the overhead for motion compensation, the syntax is optimized for such scenarios. A BI frame is not used as a reference frame. Skipped P frame is signaled when the frame is exactly the same as the previous reference.

The Intra MBs in VC-1 do not use prediction and use fixed size transform of size 8x8. The Inter-coded MBs (such as in P/ B frames) can use up to four different transform sizes for the residual: 8x8, 4x8, 8x4, and 4x4. Transform block size can change adaptively in P/ B frames with 4 different size options, while size of motion compensation is either 16x16 or 8x8 in VC-1. H.264 on the other hand uses a fixed sized transform and a variable block size for motion compensation. The transforms are 16 bit transforms where both the sums and the products of two 16 bit values produces results within 16 bits – the inverse transform can be implemented in 16 bit fixed point arithmetic. Note that the transform approximates a DCT, and norms of basis function between transforms are identical to enable the same quantization scheme through various transform types.

VC-1 supports a few options for motion compensation: 1) Half-pel or quarter-pel resolution motion compensation can be used. 2) Bi-cubic or bi-linear filter can be used for the interpolation. 3) 16x16 or 8x8 block size can be used. Only some combinations of such options are defined to signal at the Frame level. We do not provide a complete overview of VC-1 features due to space considerations. A good overview of VC-1 can be found in [9, 10]. A high level comparison of VC-1 and H.264 features is shown in Table 1.

TABLE 1: COMPARISON OF VC-1 AND H.264 FEATURES TRANSCODING

Feature	VC-1	H.264
Picture type	I, P, B, BI, Skip	I, P, B, SI, SP
Transform size	Adaptive	Fixed (baseline)
Transform	Integer, similar to DCT—4 different transform sizes	Integer, similar to DCT – 4x4 and 8x8 transforms
Intra prediction	None	Directional predictors
Motion comp.	16x16, 8x8	7 variable block sizes
Reference frames	Max 2	Max 16 (each dir.)

3. VC-1 TO H.264 INTER MODE MAPPING

The proposed MB mode mapping is for a pixel domain transcoder where the VC-1 video is fully decoded and then encoded using a H.264 encoder. The data gathered during the VC-1 decoding stage is used to accelerate the H.264 encoding stage. We assume that the VC-1 encoded bitstreams were generated with an R-D optimized encoder. The Inter MB coding in VC-1 and H.264 differ significantly. VC-1 uses a maximum of two reference frames and motion compensation is performed on 16x16 or 8x8 blocks. H.264 on the other hand can have up to 16 references frames (on each direction) and variable block sizes for motion compensation. The variable transform size used in VC-1 could offer hints on the size of the block necessary for motion compensation in H.264. In this paper, we focus on P frame transcoding with the goal of transcoding VC-1 to H.264 at baseline profile.

3.1 Mixed MB Mode Mapping

The inter MBs in the P pictures in VC-1 can have up to three 8x8 sub-blocks coded as Intra. This is a mixed mode intra case and is mapped to intra MB in H.264 since the H.264 intra mode has a reasonably good performance. Based on these observations, intra mapping is done as shown in Table 2. The details of Intra MB transcoding are reported in [11].

TABLE 2: VC-1 AND H.264 MIXED MB MODE MAPPING

VC-1 Inter MB (at least 1 8x8 Intra)	H.264 Intra Mode
3 8x8 Intra blocks	*If* (8x8 size transform used in Inter block): Intra 16x16 *Else*: Intra 4x4
2 8x8 Intra blocks	*If* (8x8 size transform used in 2 Inter blocks): Intra 16x16 *Else*: Intra 4x4
1 8x8 Intra block	*If* (8x8 size transform used in 3 Inter blocks): Intra 16x16 *Else*: Intra 4x4

3.2 Inter MB Mode Mapping

An Inter coded MB in the incoming VC-1 bitstream is coded as Inter MB in H.264. The Inter MBs in VC-1 have 2 motion compensation modes – 1 MV mode and a 4 MV mode. The 1MV mode is usually selected in VC-1 for areas that are relatively uniform and will be mapped to Inter 16x16, 16x8 or 8x16 MB in H.264 using the VC-1 transform size as a measure of homogeneity in the block to be able to differentiate the three different block sizes. Table 3 shows the proposed sub partition.

TABLE 3: VC-1 AND H.264 INTER MB MODE MAPPING

VC-1 Transform size	H.264 Inter Mode
8x8	16x16
4x8	8x16
8x4	16x8

The 4MV mode is usually selected in VC-1 for areas that have non-uniform motion. The 16x16, 16x8, and 8x16 modes are eliminated for such non-uniform MBs. The MB is then mapped to one of the other variable block sizes allowed for H.264 motion compensation based on the transform size used in VC-1. Table 4 shows the proposed H.264 sub-partition modes based on VC-1 transform size. The shape of the transform used to code the MC residual in VC-1 indicates continuous regions and performing motion compensation with that block size is likely to find a better match and improve the prediction.

TABLE 4: VC-1 AND H.264 INTER MB MODE MAPPING

VC-1 Transform size	H.264 Inter Mode
8x8	8x8
4x8	4x8
8x4	8x4
4x4	4x4

To improve the performance of H.264, a skipped MB type was added to reduce the number of operations needed to code the video stream, Skip MBs represent the case in which the motion vector can be extrapolated from its neighbors and the prediction error is zero. VC-1 doesn't have this MB type so to achieve the same complexity reduction and to be able to map to Skip MB in H.264 the two following operations are performed. First if the frame in VC-1 is a Skip frame all MB in the frame will be mapped to Skip MB in H.264. Second, the motion vectors for the neighboring MBs are inspected, if they are the same as than the MB being coded, the MB will be mapped to Skip MB in H.264.

Once the MB coding mode is mapped, the next step is to determine the motion vectors for the MB. For Inter 16x16 and the VC-1 4MV mode mapped to 4 8x8 blocks in H.264, the motion vectors are used without any refinement. For other block sizes, the VC-1 motion vector is used as a seed and the motion vectors are refined with a 5x5 refinement window. The reference frames used in VC-1 are also selected as references in H.264.

4. RESULTS AND DISCUSSION

In this section we present the experimental evaluation of the proposed low complexity transcoder and compare the quality of the video compressed with the reference transcoder (reference transcoder has full VC-1 decoding followed by full H.264 encoding without any mode mapping or MV reuse). The H.264/AVC reference software used is JM12.1. The experiments were conducted with four video sequences: "Windows Logo" 128x96, "Foreman" 176x144, "Claire" 176x144 and "Walk" 176x144. The sequences represent a wide variety of possible outcomes with slow and fast motion, fast changing backgrounds, static backgrounds and other characteristics that will normally be found in videos. The sequences are encoded at 30 fps. The H.264 encoder is setup to use the baseline profile. To run these experiments a Dell Inspiron 300m running at 1.2 GHz and 256 MB memory was used.

4.1 Mode Mapping Accuracy

Inter MB mapping is based on the size of the VC-1 transform used as described in section 3. Figure 1 shows the total percentage of MBs correctly predicted by the transcoder for the whole video sequence with different values of quantization step (QP). Note also that the definition of "correct" MB used in this paper indicates that the transcoding prediction fully matches what H.264 encoder would choose. When QP exceeds 28 this percentage remains almost constant, probably due to the fact that after certain QP the MBs type doesn't change that much as larger quantizer removes the detail in the image. The mode mapping accuracy drops at lower QP because this approach does not take the QP into consideration. For sequences with less movement (logo and Claire) the percentage of correctly predicted MBs is higher, this is because this kind of sequences contain more "skip" MBs and these MBs are correctly predicted with our algorithm. The accuracy can be improved by correcting the mode mapping for lower QPs when smaller MB partitions are more likely.

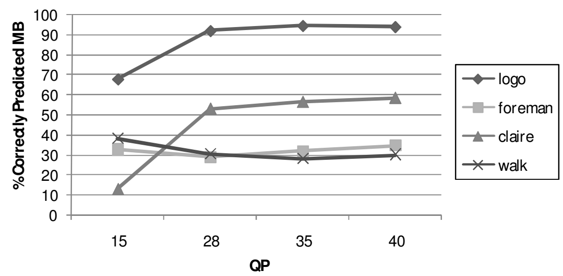


Figure 1. MB modes matching accuracy

TABLE 5: TOTAL PSNR FOR THE DIFFERENT SEQUENCES

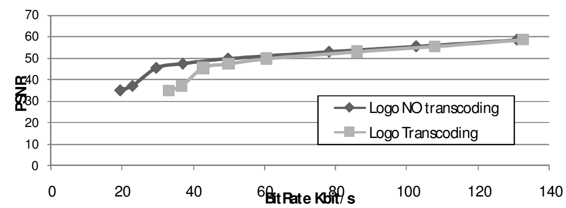
Seq.	Reference Transcoder	Proposed Transcoder
Logo	47.42	47.42
Foreman	35.35	35.29
Claire	39.57	39.52
Walk	34.88	34.84

4.1 RD Performance

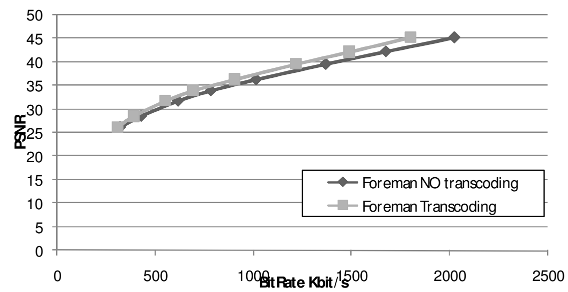
Form the RD curves in Fig. 2 we can see that the actual PSNR for the transcoded version is almost the same which confirms the objective of this paper of providing a robust transcoding technique that does not affect the PSNR significantly. The difference at higher bitrate is likely because of incorrect mode matching that does not take lower QP in to account. The subjective quality of the video sequence was also evaluated. It was noted that there is some lost of visual quality especially in areas that require more details such as the regions around the eye; this is due to errors in the Motion Vector (MV) prediction, since we are using the MV form the VC-1 decoder. The refinement of these techniques for MV prediction will be studied in future research.

TABLE 6: TIME TAKEN BY THE H.264 ENCODING STAGE

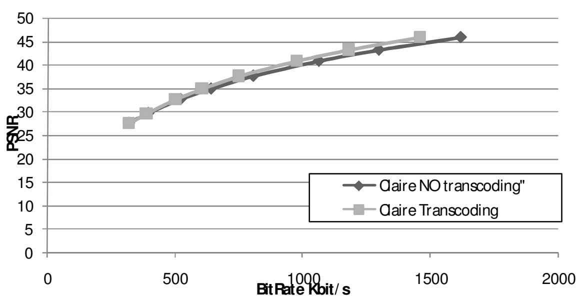
Sequence	Reference	Proposed	% Reduction
Logo	12.63	4	68%
Foreman	16.98	7	59%
Claire	16.01	7	56%
Walk	17.27	7.25	58%



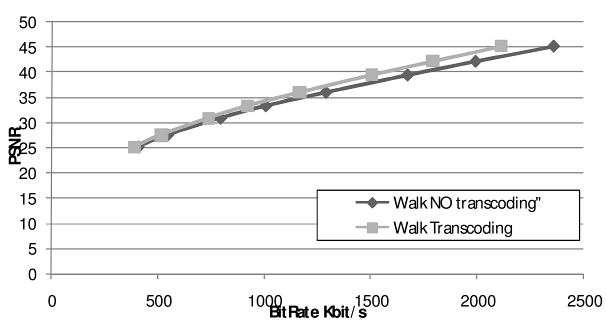
(a) Logo



(b) Foreman



(c) Claire



(d) Walk

Figure 2. RD Performance of the proposed and reference transcoders

Table 6 shows the time in seconds used to encode the different sequences by using only H.264 (reference encoder) and then using the proposed transcoder model. We can see from the table that by using the proposed transcoding algorithms we can reduce the time used to encode a sequence by around 60% percent without a significant loss in PSNR.

4. CONCLUSION

This paper addresses an important problem of transcoding VC-1 to H.264 coding format. Both VC-1 and H.264 are hybrid video coding algorithms that exploit motion compensation and transform coding. The inter MB coding in VC-1 differs significantly compared to H.264; however, VC-1 has features that can be exploited for transcoding to H.264. The proposed low complexity transcoder is based on exploiting the variable transform size used in VC-1 to determine MB coding mode in H.264. The results show that this approach works reasonable well by reducing the complexity by about 60% with negligible drop in PSNR.

5. REFERENCES

[1] J. Bialkowski, A. Kaup, K. Illgner, "Overview of Low-Complexity Video Transcoding from H.263 to H.264," Proceedings of the IEEE International Conference on Multimedia & Expo (ICME) 2006, pp. 49-52.

[2] Y.-K. Lee, S.-S. Lee, and Y.-L. Lee, "MPEG-4 to H.264 Transcoding using Macroblock Statistics," Proceedings of the IEEE International Conference on Multimedia & Expo (ICME) 2006, pp. 57-60.

[3] B. Petljanski and H. Kalva, "DCT domain intra MB mode decision for MPEG-2 to H.264 transcoding" Proceedings of the ICCE 2006. January 2006. pp. 419-420.

[4] C. Chen, P-H.Wu, and H. Chen, "MPEG-2 to H.264 transcoding," Picture Coding Symposium, 15-17 Dec, 2004.

[5] J. Xin, A. Vetro, and H. Sun, "Converting DCT coefficients to H.264/AVC transform coefficients," Proceedings of the IEEE Pacific-Rim Conference on Multimedia (PCM), IEEE PCM 2004, November 2004.

[6]Z. Zhou, S. Sun, S. Lei and M-T. Sun, "Motion Information and Coding Mode Reuse for MPEG-2 to H.264 Transcoding", Proceedings of the ISCAS 2005, May 2005. pp 1230-1233.

[7] J. Xin, A. Vetro, S. Sekiguchi, and K. Sugimoto, "MPEG-2 to H.264/AVC Transcoding for Efficient Storage of Broadcast Video Bitstreams," International Conference on Consumer Electronics, 2006. pp. 417- 418, Jan. 2006.

[8] G. Fernandez-Escribano, H. Kalva, P. Cuenca, and L. Orozco-Barbosa, "Very Low Complexity MPEG-2 to H.264 Transcoding Using Machine Learning," Proceedings of the ACM Multimedia 2007, Santa Barbara, CA, October 2006, pp. 931-940.

[9] S. Srinivassan, P. Hsu, T Holcomb, K. Mukerjee, S. Regunathan, B. Lin, J. Liang, M.-C. Lee and J. Ribas-Corbera, "WMV-9: Overview and Applications," Signal Processing Image Communication, Oct. 2004, pp851-875.

[10] SMPTE, "Standard for Television: VC-1 Compressed Video Bitstream Format and Decoding Process," SMPTE 421M-2006.

[11] J.B. Lee and H. Kalva, "An Efficient Algorithm for VC-1 to H.264 Transcoding in Progressive Video Compression," Proceedings of the IEEE International Conference on Multimedia & Expo (ICME) 2006, pp. 53-56.