3.4-4

Reduced Complexity H.264 to MPEG-2 Transcoder

Philip Kunzelmann and Hari Kalva, *Senior Member, IEEE*

Abstract—**This paper presents techniques for reducing the complexity of H.264/MPEG-2 transcoding through reuse of information gathered during the H.264 decoding stage. The H.264 macro block (MB) coding modes and the motion vectors are used for mode mapping and motion estimation in MPEG-2. We present a reduced complexity transcoder based on MB mode mapping, dynamic search range, and a dynamic search window. The proposed transcoder reduces the transcoding time by over 60% compared with the baseline cascaded transcoder with a negligible loss in PSNR.**

I. INTRODUCTION

The H.264 specification represents a significant advance in the state of video coding technology by providing video of MPEG-2 comparable quality at an average of half the required bandwidth. Although widespread adaptation to H.264 is anticipated, many legacy systems including virtually all existing digital TVs and home receivers use MPEG-2. This motivates the need for an architecture that both efficiently leverages the lower cost of H.264 video and does not require a significant investment in additional video coding hardware. Figure 1 shows the configuration of a system where such transcoders can be employed. The H.264 coded video can be delivered over IP networks and can be transcoded at the head end of a Cable TV network for delivery as MPEG-2 video. Alternatively, the H.264 video is downloaded to the set-top-box or digital video recorder (DVR) where it is transcoded for playback in MPEG-2 on the existing digital TV or DVR.

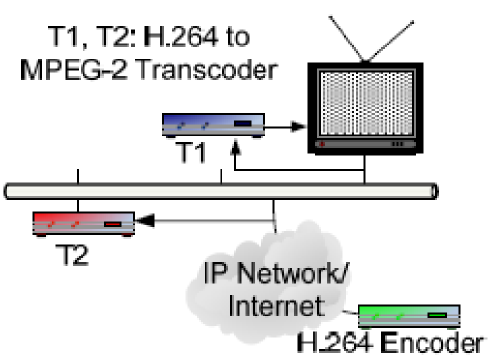

Figure 1. H.264 to MPEG-2 Transcoder Applications

Transcoding H.264, MPEG-4 and MPEG-2 to H.264 has been reported in literature [1,2,3]. All these approached to transcoding attempt to reuse the information gathered in the decoding stage to reduce the complexity of transcoding with varying performance improvements. The present problem of transcoding H.264 to MPEG-2 is different and the existing methods cannot be directly applied. Transcoding H.264 to MPEG-2 is reported in [4]. This work directly derives MPEG-2 motion vectors from the H.264 motion vectors but results in PSNR loss of over 3 dB.

Reuse of H.264 motion information for mPEG-2 encoding, however, is problematic given the advanced features of H.264. Variable block-size motion compensation and increased pixel search accuracy in particular make a direct translation of .264 motion vectors difficult and are prone to introducing error. An accurate translation becomes more difficult as the number of sub-blocks for a particular 16x16 MB increases. Each additional sub-block introduces the possibility a motion vector was found in a different direction. Using an average of these vectors to arrive at a single 16x16 vector becomes less reliable as the mean angle between the vectors increases. The proposed transcoder addresses this problem by using three approaches: 1) MB mode mapping, 2) Dynamic search range and, 3) Dynamic window. These approaches require more computation than direct motion vector mapping but the resulting drop in PSNR is negligible at less than 0.3 dB in the worst case.

II. EXPERIMENTS AND ANALYSIS

Each encoded MB has an associated coding mode. The MB mode indicates whether a MB is coded as Intra (i.e. without temporal prediction) or Inter (i.e. with temporal prediction). Mapping of H.264 MB modes to MPEG-2 may be performed to determine coding modes and avoid ME searches in areas that are more spatially related. However, as H.264 supports more encoding modes than MPEG-2, mode mapping must carefully consider which modes to map. The method used for the proposed transcoder segments H.264 MB modes into one of three categories: Skipped (MBs skipped by H.264), Inter (temporal) and Intra (spatial). Mode mapping allows costly ME time to be avoided for area where a spatial search is more appropriate. Table 1 compares encoding times for MPEG-2 reference software without changes (baseline) and the same software where mode mapping has been implemented.

Mode mapping alone, however, makes no use of H.264 motion data for inter-coded macroblocks. While it is highly desirable to apply H.264 motion vectors to MPEG-2, the

1-4244-0763-X/07/$20.00 ©2007 IEEE

increased pixel accuracy and sub-macroblock motion vectors of H.264 make direct reuse difficult. H.264 motion vectors specify a position in ¼ pixel increments where MPEG-2 uses ½ pixel increments. More problematic are the number of motion vectors encoded for a 16x16 macroblock. While MPEG-2 uses a full-search algorithm for the entire 16x16 space, an H.264 coded block may contain up to sixteen sub-blocks and their corresponding motion vectors. Even at a 16x8-only level, experimental results [4] show increased signal degradation (1 dB ~ 3 dB) where a simple mean is applied in translating motion vectors. For this reason a dynamic MPEG-2 search range is used to improve the motion vector accuracy. The search range is set equal to the size of the largest H.264 motion vector for that macro block. Table 2 compares encoding times for baseline MPEG-2 and an implementation of mode mapping and a dynamic search range. For this simulation, the search range for each macroblock is specified by the absolute value of the H.264MVs:

MPEG-2 MV Range x = Max (ABS(mvx))
MPEG-2 MV Range y = Max (ABS(mvy))

III. EXPERIMENTAL SETUP

Simulations run for the following experiments use JM 10.2 H.264 and MPEG-2 reference software from the MPEG Software Simulation Group. Each video sequence used was encoded with the following configuration: 150 frames, 720x480 resolution, YUV 4:2:0 chroma format, GOP size of 15 with I and P frames only. The H.264 input was encoded with QP of 22 (giving a PSNR of around 43 dB) and the MPEG-2 output was generated for seven different bitrates from 6 Mbps to 12 Mbps. PSNR was calculated from the decoded H.264 and reconstructed MPEG-2 YUV frames. Transcoder complexity was measured as total time spent for the MPEG-2 encoded process only (as this was the only variable). Elapsed time was measured in milliseconds and all sequences were encoded using a DELL Optiplex GX620 with a 3.0 GHz Intel Pentium D and 1 GB of RAM.

IV. CONCLUSIONS

The proposed transcoder reduces the complexity substantially with negligible loss in PSNR. Simulations run for several common video sequences (flower, hook, mobile, tennis) show mode mapping of H.264 block-types to MPEG-2 yields a decrease in encoding time anywhere from 4% to 27% depending on the sequence and target bitrate (Table 1). Signal loss introduced by mode mapping is negligible (greatest PSNR change was -0.3 dB for 'mobile') and actually improved by +0.2 dB for the test sequence 'hook'. Sequences with greater spatial consistency are likely to benefit more from mode mapping because of reduced ME complexity. Using a dynamic search range yielded a significant improvement (60% to 70%) in total encoding time. PSNR change introduced by this modification is negligible and is also in the range of -0.3 dB to +0.2 dB. The complexity can be further reduced by using a dynamic search window where the MV refinement window for MPEG-2 motion vectors is varied dynamically.

REFERENCES

[1] J. Bialkowski, A. Kaup, K. Illgner, "Fast transcoding of intra frames between H.264 and H.264," International Conference on Image Processing, 004. ICIP '04. vol. 4, pp. 2785-2788.

[2] Y.-K. Lee, S.-S. Lee, and Y.-L. Lee, "MPEG-4 to H.264 Transcoding using Macroblock Statistics," Proceedings of the ICME 2006, Toronto, Canada, July 2006.

[3] G. Fernandez-Escribano, H. Kalva, P. Cuenca, and L. Orozco-Barbosa, "Speeding-up the Macro block Partition Mode Decision for MPEG-2 to H.264 transcoding," Proceeding of ICIP 2006, Atlanta, September 2006.

[4] J. Chu *et. al*., "H.264/MPEG-2 Transcoding based on Personal Video Recorder Platform," Proceedings of the Ninth International Symosium on Consumer Electronics pp. 438-440, June 2005.

Table 1. MPEG-2 w/ Mode Mapping Encoding Times

	Total Time (sec.)							Avg. Time (baseline)	Avg. Time	Avg. Gain
Bitrate (Mbps)	6	7	8	9	10	11	12			
flower	65.2	65.4	65.8	65.8	66.0	68.2	68.3	68.6	65.8	4.15%
hook	49.3	49.5	49.7	49.8	50.0	50.2	50.3	66.5	49.8	25.04%
mobile	69.8	69.9	70.1	70.3	70.5	70.6	70.9	74.1	70.3	5.11%
tennis	91.2	91.3	91.6	91.8	91.9	92.1	92.3	100.0	91.8	8.27%

Table 2. MPEG-2 w/ Dynamic Range Encoding Times

	Total Time (sec.)							Avg. Time (baseline)	Avg. Time	Avg. Gain
Bitrate (Mbps)	6	7	8	9	10	11	12			
flower	26.8	27.0	27.3	27.4	27.6	27.8	28.0	68.6	27.4	60.04%
hook	25.8	26.0	26.1	26.3	26.5	26.6	26.8	66.5	26.3	60.45%
mobile	25.7	25.9	26.1	26.3	26.5	26.6	26.8	74.1	26.3	64.54%
tennis	30.0	30.1	30.3	30.5	30.7	30.9	31.1	100.0	30.5	69.48%

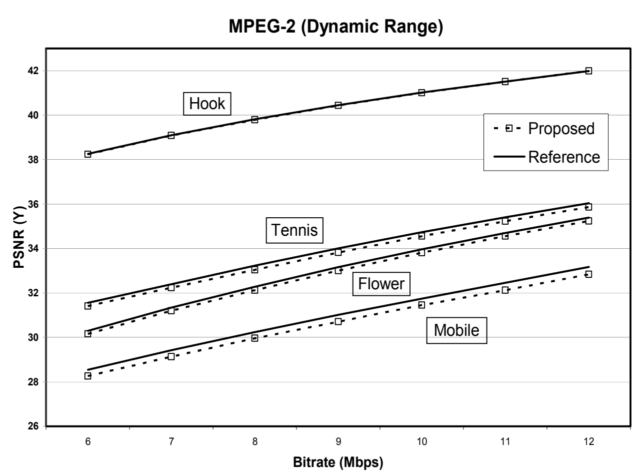

Figure 2. MPEG-2 w. Dynamic Range PSNR vs. Bitrate Comparison