Motion and Mode Mapping for MPEG-2 to H.264/AVC Transcoding

Jun Xin[1], Anthony Vetro[1], Shun-ichi Sekiguchi[2], Kazuo Sugimoto[2]

[1] Mitsubishi Electric Research Labs, Cambridge, USA
[2] Mitsubishi Electric Corporation, Kanagawa, Japan

Abstract — *This paper describes novel low-complexity transcoding techniques for an efficient MPEG-2 to H.264/AVC transcoder. We present a motion and mode mapping algorithm that directly maps incoming MPEG-2 motion information to H.264/AVC motion vectors as well as coding modes. Refinement strategies are also presented to improve performance. Experiment results demonstrate the effectiveness of the proposed low-complexity transcoder.*

I. INTRODUCTION

MPEG-2 is the primary format for broadcast video, where the data rate for high-definition video is approximately 18Mbps. The latest video coding standard, referred to as H.264/AVC [1], promises the same quality as MPEG-2 with about half the data rate. Since the H.264/AVC format has been adopted into storage format standards, such as Blu-ray Disc, we expect H.264/AVC decoders to appear in consumer HDD systems soon. Certainly, as more high-definition content becomes available and the desire to store more content or record more channels simultaneously increases, long recording mode will be a key selling point for future HDD recorders.

To satisfy this need, we have developed novel techniques that convert MPEG-2 broadcast video to the more compact H.264/AVC format with low complexity. Complexity is kept low by reusing information contained within the MPEG-2 video stream. At the same time, high quality is maintained. The diagram of the proposed system is shown in Figure 1. Since an MPEG-2 decoder is present in existing systems, the challenge is integrating the simplified H.264/AVC encoding part of the MPEG-2 to H.264/AVC transcoder into the overall system. This paper focuses on the motion and mode mapping algorithms, which are the main obstacles in a low-complexity transcoder design. Our target output is H.264/AVC baseline profile and we assume the input is MPEG-2 frame coded pictures, which is the more popular MPEG-2 coding method. To simplify the explanation of the proposed algorithms, we consider only frame predictions in frame pictures. However, the proposed method could easily be generalized for field predictions and field pictures.

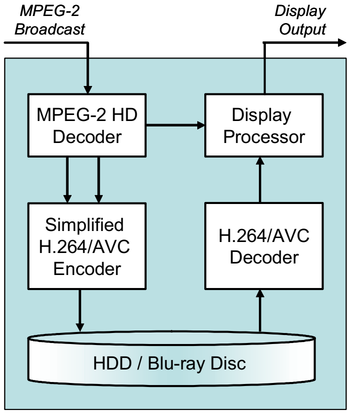

Figure 1. Storage system using MPEG-2 to H.264/AVC transcoding.

The remainder of this paper is organized as follows. First, we describe proposed low-complexity transcoding techniques. Then, we present experimental results demonstrating the effectiveness of the proposed algorithms. Finally, concluding remarks are provided.

II. TRANSCODING TECHNIQUES

Straightforward cascading of an MPEG-2 decoder and a standalone H.264/AVC encoder would form a transcoder; this will be referred to as the reference transcoder later on in this paper. The reference transcoder is very computationally complex due to the need to perform motion estimation and mode decision in the H.264/AVC encoder.

It is well known that we could reduce the complexity of the reference transcoder by reusing the motion and mode information from the input MPEG-2 video bitstream [2][3] and various algorithms to achieve this have been reported in literature [4][5]. However, it is often difficult to evaluate the performances of these approaches in terms of their complexity and rate-distortion (RD) tradeoff. In this work, we provide a relatively simple transcoder design that uses only a small subset of coding tools available in the H.264/AVC standard. One goal of this paper is to provide a lower bound on RD performance and complexity that could assist with future transcoder development. Our second goal

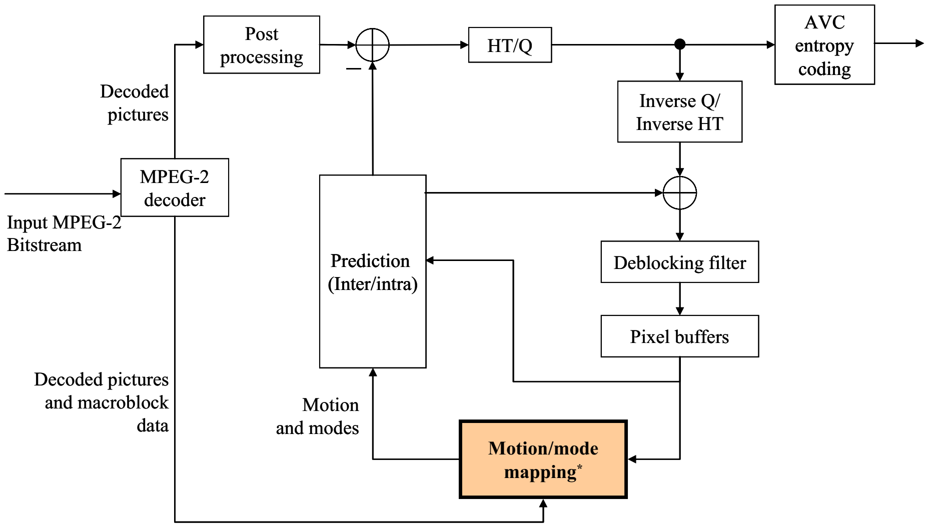

Figure 2. MPEG-2 to H.264/AVC transcoding architecture including motion and mode mapping algorithms.

is to demonstrate that satisfactory RD performance could be obtained with a small subset of coding tools and modes.

A. Transcoder architecture

The transcoder architecture is shown in Figure 2, which essentially consists of an MPEG-2 decoder, a simplified H.264/AVC encoder and a post-processing unit (following the MPEG-2 decoder) that may perform artifact removal or resolution scaling. In this paper, we are mainly interested in the motion and mode mapping block of the simplified H.264/AVC encoder. The encoder is "simplified" relative to the reference transcoder, since the motion and mode information are directly mapped from input MPEG-2 video. In this work, we emphasize on simplicity, which is the priority in making many of the following algorithm design tradeoffs.

In our system, we transcode I-frames to I-slices (pictures). Since the incoming MPEG-2 video typically contains B-frames, and B-slices are not supported in the H.264/AVC baseline profile, we must convert both P-frames and B-frames to P-slices. Given this restriction, it is necessary to change the input motion vector to match the target prediction distance as well as prediction type.

B. Mode and motion mapping

The mapping of macroblock coding modes is performed as follows.

- MPEG-2 intra mode is mapped to intra mode in H.264/AVC. H.264/AVC intra mode decision is performed, and no inter coding modes are considered.
- In B to P mapping, intra modes are always evaluated for every macroblock. We do this to avoid poor inter prediction when there are uncovered background, since those areas tend to be well predicted using backward motion, but not forward motion.
- MPEG-2 inter mode (including "skipped") is mapped to inter mode in H.264/AVC. Considering that our target application output bit rate is relatively low, we opt to test only two inter prediction modes: inter16x16 and skipped. The motion vector derivation for the inter16x16 prediction is specified below.

Next, we consider the mapping of P-frame motion vectors as shown in Figure 3. We take the first incoming P-frame as an example. In the input video, the first P-frame is predicted from its preceding I-frame. Let *inDistance* be the temporal distance between the P-frame and its reference I-frame. In this example, *inDistance=3*. Assuming one temporal reference frame in the output video, the set of input motion vectors must be modified to reference the preceding P-frame in the output H.264/AVC video. Denote the temporal distance between the output P-picture and its reference P-picture as *outDistance*. In this example, *outDistance=1*. Assuming the motion is small and linear in the period of *inDistance* frames, which is typically 100ms or less, we can represent the mapping from the motion vector from the incoming MB to the output MB with:

$$outputMV = (inputMV \div inDistance) \times outDistance \quad (1)$$

If the incoming macroblock is skipped, the input motion vector is set to 0.

In B to P motion mapping, it is more complicated since the input picture contains both forward and backward motion vectors. We consider the example illustrated in Figure 4, i.e. mapping the second incoming B-frame to the second outgoing P-picture.

- For the case when a macroblock in the B-frame has a forward motion vector (*mvForw*), we use equation

Table 1. Complexity of the reference transcoder and the proposed (baseline+refinement) transcoder for Crew and BigShips sequences. Time is measured in seconds.

Quality Setting	Transcode Method	Trans. Time [Crew]	Trans. Time [BigShips]
QP=26	Reference	5272	5006
	Proposed	342	322
	Savings	***93.5%***	***93.6%***
QP=35	Reference	9575	7271
	Proposed	320	388
	Savings	***96.7%***	***94.7%***

(1) to calculate the output motion vector for the P-picture by scaling the incoming forward motion vector *mvForw*. Here we again assume that the motion between these frames is small and linear.

- For the case when a macroblock in the B-frame has only a backward motion vector (*mvBack*), we first add the backward motion vector to the forward motion vector of the collocated future P-frame, i.e., *mvInt* = (*mvBack* + *mvCol*). We then scale the resulting motion vector according to (1), where *mvInt* is an intermediate motion vector that is used in (1) as input.
- For the case when the incoming macroblock in the B-frame is skipped, the input motion vectors are set to be the motion vectors that would be used for decoding the skipped macroblock.

C. *Motion vector refinement*

As we will see in the simulation results, the motion vectors obtained through the simple motion mapping algorithms described above are coarse and need refinement to approach the performance of the reference transcoder.

The refinement strategy used in our transcoder system is as follows. The motion vector mapping results are first rounded to nearest integers. Then, we perform a 3x3 window of integer pixel refinement around the mapped motion vector, followed by a 3x3 window of half-pixel refinement around the best integer motion vector. Finally we perform a 3x3 window of quarter-pixel refinement around the best half-pixel motion vector.

III. SIMULATION RESULTS

In our experiments, we use 120 frames of the BigShips and Crew sequences, which have a resolution of 1280x720p and frame rate of 60fps. The sequences are encoded to a bit-rate of 18Mbps with MPEG-2 using a GOP structure of N=30, M=3. The target output is H.264/AVC baseline profile with a single reference frame. The set of fixed QP values used for encoding are: {26, 29, 32, 35}.

In the simulations, we compare three transcoders:

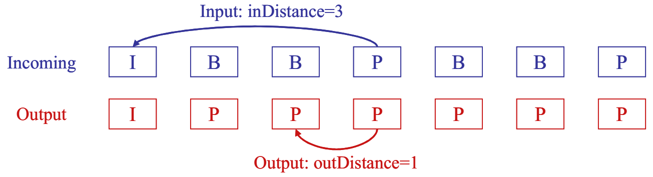

Figure 3. Motion vector mapping of P-frames.

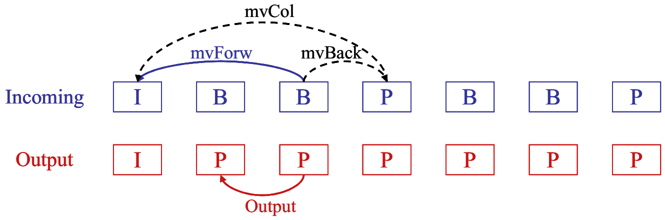

Figure 4. Motion vector mapping of B-frames.

- Reference transcoder: cascaded decoder and the standalone H.264/AVC encoder (JM7.6) using baseline profile tools. The H.264/AVC encoder uses full motion estimation with quarter-pixel accuracy and exhaustive mode decision. The motion estimation search range is (-64, 64) and RDO is set to off.
- Baseline transcoder: using motion and mode mapping algorithms as described in section II, but without any motion refinement.
- Baseline transcoder: using motion and mode mapping algorithms as described in section II, but with the described motion refinement.

As shown in Figures 5 and 6, the baseline+refinement transcoder performs nearly as well as the reference transcoder for BigShips, and is less than 0.5dB below the reference for Crew at the bit-rate range of interest (6~8Mbps). Another notable observation is the amount of improvement that only a small range of motion vector refinement provides. These simulation show that PSNR gains between 1.5dB and 3dB could be obtained depending on the bit-rate.

In terms of computational complexity, our proposed baseline+refinement transcoder uses less than 7% of the time used by the reference transcoder. Table 1 shows the savings achieved by the proposed transcoder relative to the reference transcoder.

IV. CONCLUDING REMARKS

We have presented a cost-effective transcoder design. The proposed transcoder incorporates simple mode and motion mapping algorithms for MPEG-2 to H.264/AVC baseline profile video transcoding. We discussed mapping the motion vectors of MPEG-2 input P- and B-frames to H.264/AVC inter motion vectors. In our transcoder, we used only two inter prediction modes: 16x16 and skipped. We performed motion vector refinement for the 16x16 inter prediction using a small refinement window (1+1/2+1/4). Simulation results showed that the proposed "baseline+refinement" transcoder performs nearly as well as the reference transcoder in terms of R-D, while it consumes less than 7% of the run-time complexity. We plan to make further comparisons with fast motion estimation in the reference and RDO enabled.

REFERENCES

[1] ITU-T Rec. H.264 | ISO/IEC 14496-10, "Advanced Video Coding," 2003.

[2] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18-29, Mar. 2003.

[3] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," Proceedings of the IEEE, vol. 93, no. 1, pp. 84-97, Jan. 2005.

[4] Z. Zhou, S. Sun, S. Lei and M.T. Sun, "Motion information and coding mode reuse for MPEG-2 to H.264 transcoding," *Proc. IEEE Int'l Symp. Circuits Syst.*, Kobe Japan, May 2005.

[5] X. Lu, A. Tourapis, P. Yin and J. Boyce, "Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding," *Proc. IEEE Int'l Symp. Circuits Syst.*, Kobe, Japan, May 2005.

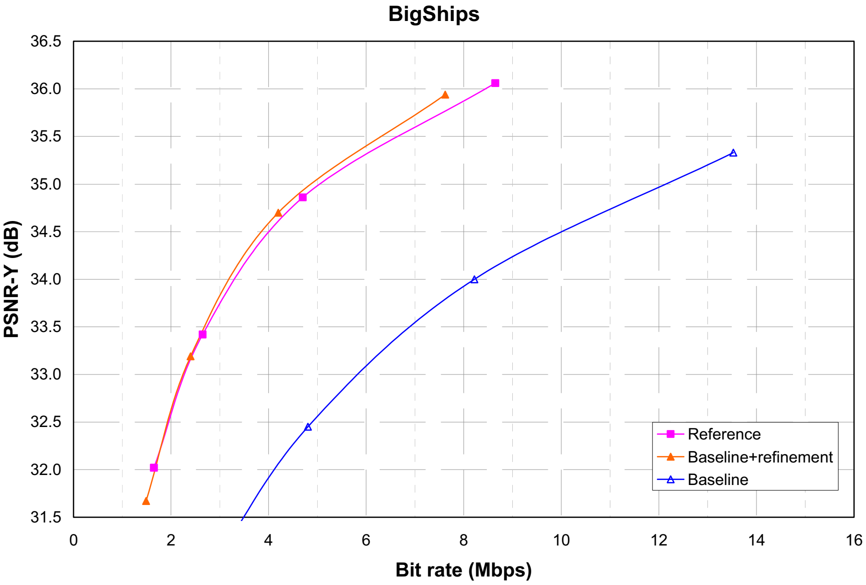

Figure 5. RD performance of the reference transcoder and the proposed transcoder

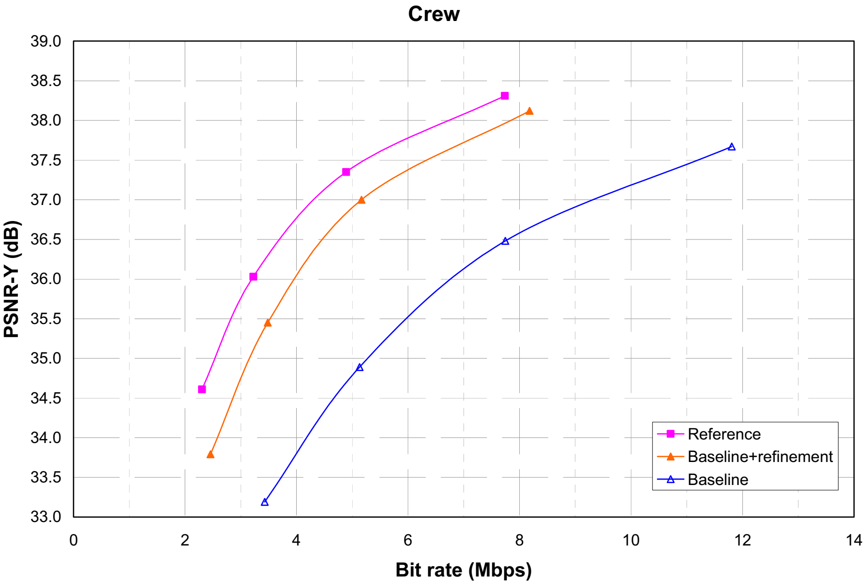

Figure 6. RD performance of the reference transcoder and the proposed transcoder