A COMPLEXITY-SCALABLE TRANSCODER FROM H.264/AVC TO THE NEW HEVC CODEC

Eduardo Peixoto and Ebroul Izquierdo

Queen Mary, University of London, London, UK
Email: eduardopeixoto@ieee.org, ebroul.izquierdo@elec.qmul.ac.uk

ABSTRACT

The emerging video coding standard, HEVC, is currently approaching the final stage of development prior to standardization. However, the current H.264/AVC standard is very successful, and it has been widely adopted for many applications. Thus, transcoding between these codecs will be highly needed once the HEVC codec is finalised. This paper studies the performance of one of the most common techniques for heterogeneous transcoding, motion vector (MV) reuse, in a H.264/AVC to HEVC transcoder. Furthermore, it proposes a new transcoder that is capable of complexity scalability, trading off rate-distortion performance for complexity reduction. The proposed transcoder is based on a new metric to compute the similarity of the H.264/AVC MVs, which is used to decide which HEVC partitions are tested on the transcoder.

Index Terms— H.264/AVC, HEVC, Transcoding

1. INTRODUCTION

The emerging video coding standard, so called High Efficient Video Coding (HEVC) [1], is being developed by the ITU-T and JCT-VC groups to replace the current H.264/AVC standard [2]. Even before it is finalised, in its recent iteration, the HEVC outperforms the H.264/AVC by 30% to 50% [3].

The H.264/AVC standard is very successful and it has been widely adopted, both by physical media, internet streaming services and some broadcast and cable television services. Hence, there will be a need for efficient conversion between the H.264/AVC standard and the HEVC, once it is finalised.

Transcoding techniques have been thoroughly studied in the literature [4, 5, 6]. Transcoding from H.264/AVC to the HEVC requires a change of format, characterizing it as an heterogeneous transcoder [4]. The most straightforward way to convert between formats is to cascade the source decoder with the target encoder, without any intermediate processing. Here, this is referred as the trivial transcoder.

Most heterogeneous transcoders follows the cascaded pixel-domain approach [4, 5], where the source bitstream is decoded, its data is processed (motion information, transform decisions, coefficients, etc...) and then used to re-encode the sequence with the target codec. One technique that is ubiquitous among heterogeneous transcoders is the motion vector (MV) reuse [7]. The rationale of this technique is to use the MVs from the source codec in the target codec, avoiding performing a costly motion estimation (ME) operation. In some works, a refinement step on these MVs is also considered in order to improve rate-distortion (RD) performance [4], but even in these cases, the refinement is generally performed using a fast ME technique or a reduced search window, so the complexity savings are kept. Since ME is the most time consuming operation in a video encoder, this technique alone can significantly reduce the transcoder complexity.

There are two main challenges in applying this technique in the H.264/AVC to HEVC transcoder. The first, and obvious one, is that the HEVC uses much larger blocks for motion estimation than the H.264/AVC (64×64 [8] against a fixed 16×16 largest unit). Thus, the motion information has to be combined and merged in order to be efficiently reused in the HEVC. The second challenge is that, in the main anchor profiles for the HEVC, full motion estimation is not used, being dropped in favor of a fast motion estimation technique that can achieve a performance close to the full motion estimation, at a much lower complexity cost, specially for HD and larger content. The reason for this change is that, even if complexity is not an issue, there are other techniques that could be used to further improve encoding performance while still using less complexity than full ME (for example, instead of using full ME, fast ME could be used with more reference frames, or the RD optimization module could be allowed to test more quantization parameters - QPs - in both cases, the complexity added would be much lower than using full ME, but the potential gain is higher).

In this paper, we discuss the performance of a modified MV Reuse technique in a H.264 to HEVC transcoder. Then, we propose a transcoder based on a new method that is capable of complexity scalability, trading off RD performance for complexity reduction. The method is based on a new metric to compute the similarity of the H.264/AVC MVs, and it uses this metric to decide which HEVC partitions are tested on the transcoder. Furthermore, the proposed transcoder is generic in the sense that its main techniques can still be used even if the HEVC changes some of its underlying algorithms.

This research was partially supported by the European Commission under contract FP7-247688 3DLife.

2. A H.264/AVC TO HEVC TRANSCODER BASED ON MV REUSE

In order to study the performance of an H.264/AVC to HEVC transcoder, a simple transcoder based on the cascaded pixel domain approach was implemented. This transcoder is mainly based on the MV reuse technique [7], and will be referred here as the MV Reuse transcoder.

Here, we define the testing of a block as the assessment of the best way to encode that particular block (i.e., deciding the parameters - MVs, transforms, etc... - and producing a RD cost, which will then be compared to decide which mode will be used to encode that block). Similarly, we define the testing of a MV as the evaluation of the cost of that MV, and comparing this cost with the MVs that were previously tested. The workflow of the algorithm is the same for each coding unit (CU) in the HEVC, and it is based on two main ideas:

1. If any part of this CU was encoded in intra mode in the H.264/AVC, then all possible intra and inter modes are tested; otherwise, only the inter modes are tested.

2. For any inter partition unit (PU), all H.264/AVC MVs within the PU are tested, without any further refinement at integer-pixel level. Then, at half-pixel and quarter-pixel, the default HEVC search is applied.

All inter modes available in the HEVC are considered (including the Assymetric Motion Partition, AMP [9] - for these partitions, the AMP speed-up setting is enabled). The remaining HEVC settings are the same as the low-delay configuration for HM 4.0rc1 [8], including the fast mode decision flag (which is enabled). For the H.264/AVC, a similar low-delay *IPPP* configuration with 4 reference frames and High Profile was used, using JM 14.0. Selected results can be seen in Fig. 1. In the tests, the QPs used are $\{37, 32, 27, 22\}$ in both codecs, and the transcoder uses the same QP as the H.264/AVC bitstream. In addition, the results for the trivial transcoder (i.e., fully decoding and re-encoding the sequence, with the HEVC settings) are also shown.

The complexity results can be found in Table 1. The trivial transcoder is used as anchor for both BD-bitrate (the average bitrate difference relative to an anchor, in percentage) [10] and complexity (measured here as the total running time, and shown as the relative speed-up). It can be seen from Fig. 1 that, as expected, there is a loss in quality in transcoding, since the transcoder does not operate on the original sequence ($-1.52dB$, $-1.72dB$ and $-1.18dB$, on average, for Basketball Drill, BQMall and Vidyo1 sequences, respectively). However, there is also a considerable bitrate reduction. As for the MV Reuse transcoder, the rate-distortion results are close to the trivial transcoder, except for Race Horses sequence, demonstrating a high correlation between the H.264/AVC and the HEVC MVs (as can be seen in the figures and in the BD-Bitrate values). However, the complexity savings are not as high as expected, ranging from a speed-up factor of 1.12 (for Vidyo1 sequence) to a maximum of 1.77 (for Race Horses sequence). The speed-up shown in Table 1 is the average between the four QPs used. Note that the trivial transcoder uses fast ME and fast mode decision.

3. THE PROPOSED TRANSCODER

The transcoding techniques proposed here aims primarily to reduce the complexity of the MV Reuse transcoder shown in Sec. 2. It achieves this mainly by testing less CUs and PUs than that transcoder. The proposed transcoder uses a similarity metric, the MV Variance Distance and, according to this metric, it makes the decision of which PUs will be tested for a given CU and whether this CU will be split or not.

The similarity metric produces a value $\upsilon \geq 0$ for each CU in the HEVC, and two thresholds T_{low} and T_{high} are used to decide how that CU will be tested. This metric is based on the variance of the MVs, and it is computed as:

$$\upsilon = \sqrt{\left(\sigma_x^2\right)^2 + \left(\sigma_y^2\right)^2} \qquad (1)$$

where σ_x^2 and σ_y^2 are the variances of each component of the MVs within the CU. If the MVs do not have the same reference frame, then the metric does not produce a value.

3.1. Transcoder Algorithm

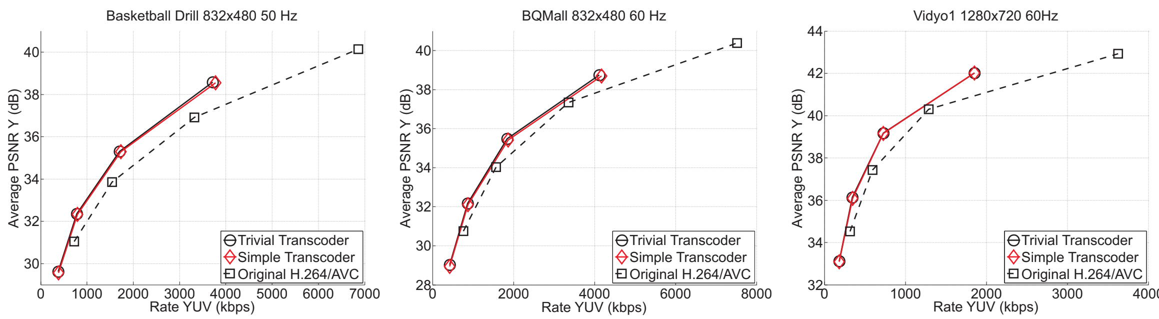
Starting from the largest size CU, the transcoder divides the possible partitions units (PUs) to be tested in 4 groups: (i) SKIP; (ii) inter $2N \times 2N$; (iii) all remaining inter modes ($2N \times N$, $N \times 2N$, the AMP modes, and $N \times N$); and (iv) the intra modes ($2N \times 2N$, $N \times N$ and PCM). In addition, the transcoder can decide if the CU will be split or not (if so, the CU is split in four sub-CUs, as usual). Then, depending on the value υ for this particular CU, three different settings can be used:

1. if the CU is considered similar (i.e., if $\upsilon \leq T_{low}$), then only the PU groups (i) and (ii) will be tested and the CU will not be split;

2. if no information can be gathered for this particular CU (i.e., if $T_{low} < \upsilon \leq T_{high}$, or if the similarity metric cannot be computed for this CU), then all PU groups will be tested and the CU will be split; and

3. if the CU is considered as dissimilar (i.e., if $\upsilon > T_{high}$), then only the PU groups (i) and (iii) will be tested, and the CU will be split.

When a CU is split, the algorithm is applied in the same manner to each of the four sub-CUs, computing a new similarity value υ for the sub-CUs, until the final possible depth is reached. Note that the SKIP mode is always tested, even if the CU is considered dissimilar, as the complexity to test the SKIP is small, compared to the other modes. Furthermore, as in the simple transcoder shown in Sec. 2, the intra modes are



(a) (b) (c)

Fig. 1. Results of the trivial transcoder and for: (a) Basketball Drill; (b) BQMall; and (c) Vidyo1.

only tested if some part of this CU was coded as intra in the H.264/AVC bitstream, and the default fast mode decision algorithms in the HEVC still applies (for testing the intra PCM and to decide for early skip).

3.2. MV Reuse and Refinement

In addition to the similarity measure, the H.264/AVC MVs are also reused in the transcoder. There are two main ways by which the MVs are reused: (i) for a given PU, the H.264/AVC MV that covers the largest area within that CU is also tested as a MV candidate (along with the $(0, 0)$ MV and the median MV), then the default HEVC fast motion search is applied; and (ii) all H.264/AVC MVs are considered for integer ME, and no further refinement is performed at the integer pixel level. In both cases, the default HEVC sub-pixel search is applied.

3.3. Using MV Scaling to compute the similarity metric

The MV Variance Distance can only produce a value if all MVs being considered at a given point share the same reference frame. However, in order to overcome this limitation and further reduce the complexity, the MVs can be scaled to the same reference frame before the similarity value is computed. Here, a simple scaling formula is used:

$$mv_{n \to n-\beta} = \left(\frac{\beta}{\alpha} \right) \cdot mv_{n \to n-\alpha} \tag{2}$$

where n is the current frame, $n - \alpha$ is the reference frame used by the H.264/AVC MV and $n - \beta$ is the target reference frame. If the scaling is necessary, then all MVs are scaled to the frame which is closest to the current frame.

4. EXPERIMENTAL RESULTS

The test settings in this section are the same as those in Sec. 2. Here, several combinations of the methods discussed in Sec 3 are evaluated, each one targeting a different complexity level. The exact parameters can be seen on Table 2. The transcoder options are named in descending order of complexity. Selected results are shown as rate-PSNR curves in Fig. 2.

Table 2. Parameters used for the proposed transcoder.

Test Name	T_{low}	T_{high}	Scaling MVs	Refinement
Proposed (i)	1	Not Used	No	Yes
Proposed (ii)	1	Not Used	Yes	Yes
Proposed (iii)	1	100	Yes	Yes
Proposed (iv)	1	100	Yes	No

As expected, by varying the value of T_{low} one can affect the complexity of the proposed transcoder. The higher the value, the lower the complexity of the proposed transcoder (as more CUs are considered similar), and the higher the impact on the RD performance (as less CUs are tested). Experiments showed that using a value of T_{low} higher than 1 generally offers small complexity savings (compared to using $T_{low} = 1$) but may have a higher impact on RD performance. Similarly, using a value of T_{high} lower than 100 generally provides small complexity savings (compared to using $T_{high} = 100$) but may have higher impact on RD performance.

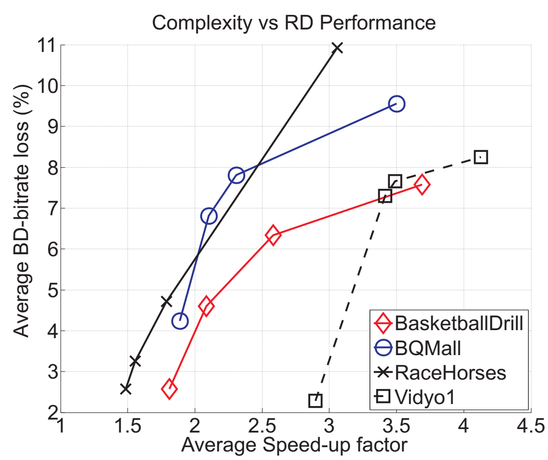
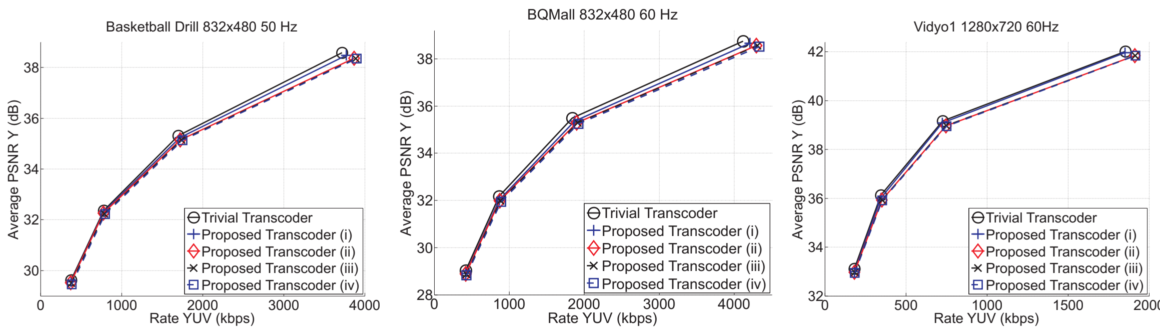


Fig. 3. Complexity and RD Performance results for the proposed transcoder for different sequences. For each sequence, each point in the curve refer to the results using a particular option for the proposed transcoder (from (i) to (iv)). In all cases, the anchor is used as the trivial transcoder (i.e., decoding and re-encoding the sequence).

Table 1. BD-Bitrate and Speed-Up results for: Basketball Drill (832 × 480, 50 Hz); BQMall (832 × 480, 60 Hz); Vidyo1 (1280 × 720), 60 Hz; and Race Horses (832 × 480, 30 Hz).

	Basketball		BQMall		Vidyo1		RaceHorses	
Method	Speed-Up	BD-Rate	Speed-Up	BD-Rate	Speed-Up	BD-Rate	Speed-Up	BD-Rate
Trivial	1.00	0.0%	1.00	0.0%	1.00	0.0%	1.00	0.0%
MV Reuse	1.40	2.78%	1.56	3.16%	1.12	0.74%	1.77	7.71%
Proposed (i)	1.81	2.57%	1.88	4.25%	2.90	2.28%	1.48	2.57%
Proposed (ii)	2.08	4.60%	2.10	6.81%	3.42	7.30%	1.56	3.25%
Proposed (iii)	2.58	6.34%	2.31	7.80%	3.49	7.66%	1.78	4.71%
Proposed (iv)	3.69	7.58%	3.50	9.55%	4.13	8.24%	3.05	10.92%



(a) (b) (c)

Fig. 2. Proposed Transcoder results for: (a) Basketball Drill; (b) BQMall; and (c) Vidyo1.

It can be seen from Fig. 2 that, for low bit-rates, all four proposed transcoders have a very good RD performance. For medium and higher bitrates, options (i) and (ii) still perform well, but options (iii) and (iv) incur in a higher loss.

Fig. 3 shows the speed-up versus the bitrate loss for various sequences. The speed-up shown is the average for the four QPs used. Again, note that the speed-up results of the transcoder are shown comparing to the trivial transcoder using fast ME and fast mode decision.

The proposed transcoder techniques are up to 4 times faster than the trivial transcoder, and up to 3.7 times faster than the MV Reuse transcoder presented in Sec. 2. The only case where the MV Reuse transcoder is faster than the proposed transcoder is for Race Horses sequence, where the it is faster than options (i) and (ii) of the proposed transcoder (18% and 13%, respectively). However, the RD performance of options (i) and (ii) is much better in this case. The options (iii) and (iv) are always faster than the MV Reuse transcoder. This complexity reduction is achieved at a loss of RD performance, however, even for the same levels of RD performance, the proposed transcoder (option (i)) is up to 2.6 faster than the MV Reuse transcoder (for Vidyo1 sequence).

5. CONCLUSION AND FUTURE WORK

An efficient transcoder from H.264/AVC to the HEVC codec was presented, capable of exchanging transcoder complexity at the cost of rate distortion performance. New techniques to efficiently map H.264/AVC motion information into HEVC CUs were presented. Even though the HEVC codec is not yet finalised, the proposed transcoder is generic in the sense that it could be easily adapted even if the HEVC changes some of its constituent algorithms.

For future work, more of the H.264/AVC information can be reused in the transcoder, such as the DCT coefficients, in order to further reduce the transcoder complexity.

6. REFERENCES

[1] "Joint Call for Proposals on Video Compression Technology", ISO/ IEC JTC1/SC29 /WG11/ N11113, ITU-T Q6/16 document, VCEG-AM91, Jan. 2010.

[2] ITU-T. Itu-t recommendation H.264, Advanced Video Coding for Generic Audio-visual Services, ITU-T, March 2005.

[3] B. Li, G. J. Sullivan and J. Xu, "Comparison of Compression Performance of HEVC Working Draft 4 with AVC High Profile", Doc. JCTVCG399, Nov. 2011.

[4] A. Vetro, C. Christopoulos and H. Sun, "Video Transcoding Architectures and Techniques: An Overview", in *IEEE Signal Proc. Mag.*, vol.20, pp. 18-29, Mar. 2003.

[5] J. Xin, C.-W. Lin and M.-T. Sun, "Digital Video Transcoding", in *Proceedings of the IEEE*, vol.93, pp. 84-97, Jan. 2005.

[6] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang. "Video transcoding: An overview of various techniques and research issues." IEEE Trans. on Multimedia, 7(5):793804, October 2005.

[7] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," IEEE Trans. on Consumer Electronics, vol. 44, no. 1, pp. 8898, February 1998.

[8] K. McCann, B. Bross, S.-I. Sekiguchi and W.-J. Han, "HM4: High Efficiency Video Coding (HEVC) Test Model 4 Encoder Description", Doc. JCTVC-F802, Jul. 2011.

[9] I.-K. Kim, W.-J. Han, J.H. Park and X. Zheng "CE2: Test results of assymetric motion partition (AMP)", Doc. JCTVC-F379, Jul. 2011.

[10] Gisle Bjontegaard, "Improvements of the BD-PSNR model", ITU-T SG16/Q6, 35th VCEG Meeting, Berlin, Germany, 16th - 18th July, 2008, Doc.VCEG-AI11.