Fast H.264/AVC to HEVC Transcoding based on Machine Learning

Eduardo Peixoto
Departamento de
Engenharia Elétrica
Universidade de Brasília
Campus Universitário Darcy Ribeiro
Email: eduardopeixoto@ieee.org

Bruno Macchiavello
Ricardo. L. de Queiroz
Departamento de Ciência da Computação
Universidade de Brasília
Campus Universitário Darcy Ribeiro
Email: bruno@image.unb.br
queiroz@ieee.org

Edson Mintsu Hung
Faculdade UnB-Gama
Universidade de Brasília
Campus Universitário Darcy Ribeiro
Email: mintsu@image.unb.br

***Abstract*—Since the HEVC codec has become an ITU-T and ISO/IEC standard, efficient transcoding from previous standards, such as the H.264/AVC, to HEVC is highly needed. In this paper, we build on our previous work with the goal to develop a faster transcoder from H.264/AVC to HEVC. The transcoder is built around an established two-stage transcoding. In the first stage, called the training stage, full re-encoding is performed while the H.264/AVC and the HEVC information are gathered. This information is then used to build a CU classification model that is used in the second stage (called the transcoding stage). The solution is tested with well-known video sequences and evaluated in terms of rate-distortion and complexity. The proposed method is 3.4 times faster, on average, than the trivial transcoder, and 1.65 times faster than a previous transcoding solution.**

I. Introduction

Nowadays, the H.264/ACV standard [1] is widely used in internet video streaming, Blu-ray discs and several HDTV broadcasts systems, including the Brazilian standard. In the past year a new video codec standard called High Efficiency Video Coding (HEVC), formally known as ITU-T H.265 [2] or ISO/IEC 23008-2 [3], was introduced by the ITU-T and JCT-VC groups. This new standard can achieve better compression rates than H.264/AVC, specifically for high resolution video. It is expected that HEVC should replace H.264/AVC in most applications in the near future. Therefore, converting H.264/AVC bitstream contents to the HEVC standard is an important issue.

Transcoding is defined as the process that converts a compressed bitstream (referred as the source bitstream) to another compressed bitstream (called the transcoded bitstream) [4], [5], [6]. There are several needs for video transcoding. A previously encoded video data may need to be transcoded in order to comply with specific network requirements, such as bitrate, resolution, frame rate, among others. In this work, however, we focus on heterogeneous video transcoding, which aims to convert the source bitstream to a different coding standard, herein from H.264/AVC to HEVC.

The trivial solution to this problem is to fully decode the source bitstream and completely re-encode it in the target codec. This procedure is usually referred as the trivial transcoder. While trivial transcoding can achieve the best results in terms of rate-distortion (RD), it is a time-consuming task. In order to speed-up the transcoding process the previously encoded information, such as the prediction modes, motion information and encoded residuals, can be used to re-encode the video data. This is even more critical in high bitrate video, due to the large amount of data involved.

In our previous works [7], [8], we proposed a transcoder that uses machine learning techniques in order to decide which HEVC modes will be tested. It uses features extracted from the H.264/AVC bitstream, such as the motion vectors and DCT coefficients, to build the machine learning model. The main contributions of this paper is a more robust feature selection for the training process, and a better CU classification strategy that targets faster transcoding.

II. Related Work

Despite of a vast literature in transcoding few transcoders target the new HEVC standard. One of the first transcoders from H.264/AVC bitstreams to the HEVC [9] is based on the power-spectrum rate-distortion optimisation (PS-RDO) method [10]. In that work, the motion vector (MV) cost in the transcoder is estimated from the MV variation and power-spectrum of the prediction signal resulting from that MV. The PS-RDO model is used both for mode mapping, to determine the HEVC CU partitioning, and for MV approximation, determining the MV used for each prediction unit (PU). In [11], another approach to speed-up transcoding to the HEVC focus on implementing an algorithm using Wave front Parallel Processing (WPP) and Single Instruction Multiple Data (SIMD) acceleration, along with expedited motion estimation (ME) and mode decision by using information extracted from the input H.264/AVC stream. Another interesting work [12] proposes an HEVC transcoder applied to video surveillance. In this work, each CU is classified in different categories: background, foreground and hybrid. Different strategies of CU partition termination, PU candidate selection and motion estimation simplification are used to reduce the complexity in the transcoding process.

In our previous works, we proposed several different transcoding strategies for the HEVC standard [13], [8], [14], [7]. Our first work used fixed thresholds to determine the partition of HEVC coding unit (CU) based on the H.264/AVC motion vectors (MVs) [13]. Although it presented a good rate-distortion performance over various sequences, it's main

This work was partially supported by CNPq under grants 302853/2011-1, 476176/2013-1 and 500370/2013-3

concern is the use of fixed thresholds, regardless of the content of the sequences or the conditions of the transcoding (such as the QP). So, in [7] we proposed a dynamic thresholding to tackle this issue.

In our most recent works , we proposed a content modeling transcoder using linear discriminant functions (LDFs), applied both to H.264/AVC to HEVC transcoding [8], [7] and to MPEG-2 to HEVC transcoding [7]. This transcoder is based on two well-defined stages: training and transcoding. During the training stage, the trivial transcoder is applied (i.e., all modes and CU sizes are tested), and both the information on how the H.264/AVC encoded each region (collected from the incoming bitstream) and on how the HEVC chose to encode it (collected directly from the HEVC decision engine) are gathered. Then, this information is used to build a model that is customised to that sequence and encoding conditions (such as the quantisation parameters, QPs, coding configuration, etc..). This model is then used in the transcoding stage to map H.264/AVC decisions into HEVC decisions. This transcoder presents a good rate-distortion performance, but it offers a limited speed-up, since several modes are still tested for each CUs. Therefore, the main goal of this work is to further reduce the transcoder complexity, while attempting to limit the rate-distortion loss.

III. The Proposed Transcoder

The transcoder proposed in this paper builds on our previous work [8], which is denoted here as RT-LDF (reference transcoder). The new proposed techniques have the purpose of speeding-up transcoding, and the key differences between the proposed transcoder and RT-LDF are highlighted.

The proposed transcoder operates in two distinct stages: *training* and *transcoding*. In this paper, we consider only one training stage, performed at the beginning of the sequence. When this stage ends, the transcoder builds a model which is then used in the transcoding stage. Other training strategies, whether repetitive (i.e., every n frames) or triggered (e.g., when a scene change is detected) could be used, but are out of the scope of this paper.

The transcoding operations are based on the HEVC CU. The decision starts at the LCU (always used as 64×64) using a top-down approach, continuing recursively for each sub-CUs. According to the CU depth, different mapping strategies are used. Since for CUs at depths 2 (16×16) and 3 (8×8) the H.264/AVC MB fits the HEVC CU size, a simple mode mapping algorithm is applied. This algorithm consists on testing the partitions that are the same size or larger as the matching H.264/AVC partition. For CUs at depths 0 and 1, the LDF mapping is used.

For all CUs,regardless if the LDF mapping is used, motion estimation is carried by a simple MV reuse algorithm. For any PU size, all H.264/AVC MVs within the area defined by the PU are considered for integer pixel ME (and only these MVs are tested at integer pixel level). At the sub-pixel level, the default sub-pixel search is applied at half and quarter-pixel levels.

A. LDF Mapping

The LDF mapping consists on using the H.264/AVC information in order to decide whether the current CU is split or not. A flowchart of this algorithm is shown in Fig. 1. First, the H.264/AVC information in the area defined by the current CU is used to compute some features, which are then fed into a LDF classifier. The classification is binary: *split* and *not split*. The main differences between the proposed transcoder and RT-LDF are in the LDF mapping, regarding both the way in which partitions are tested, the way that intra H.264/AVC MBs are handled and the LDF model used.

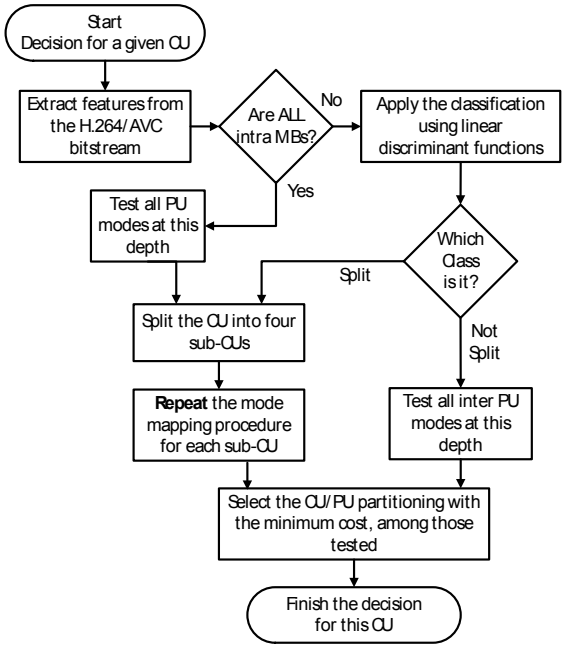

Fig. 1. Flowchart of the CU Decision Algorithm.

During the transcoding stage, the LDF classification is not used only if all H.264/AVC MBs in the current CU were encoded as intra. Otherwise, the LDF classification is used (i.e., even if there are intra H.264/AVC MBs in the current CU). This is different from RT-LDF, where the LDF classification is only used if there are no intra H.264/AVC MBs in the current CU. In order to achieve this, the way some features are computed was modified, which will be explained in the following section.

When a given HEVC CU is classified as *not split*, then all HEVC inter modes at this depth are tested, including the SKIP/MERGE mode, and the best mode, in rate-distortion sense, is chosen. The transcoder then moves to the next CU. Otherwise, if an HEVC CU is classified as *not split*, then no HEVC mode is tested at this depth, the CU is split and the transcoding decision continues recursively for the sub-CUs at the next depth. This is differently from RT-LDF, where the SKIP/MERGE is always tested, even if the CU is classified as *split*.

B. LDF Model

As in our previous works [8], [7], a simple machine learning algorithm is used, the linear discriminant functions [15]. The main reason for this choice is that, since the model is built online (i.e., during transcoding), we need to use an algorithm that presents a low complexity training, while still showing a good performance.

The two most common coding configurations used in the HEVC are the low-delay and the random access configurations. Both of these configurations encode subsequent frames with different QPs. In these coding configurations, a given base QP is used to encode only the intra frames, and subsequent inter frames are encoded using a QP offset of $\{+3, +2, +3, +1\}$. This changes the typical RD cost of the CUs in each frame, significantly changing the way the HEVC mode decision engine works. In particular, the mode distribution is greatly affected - CUs encoded with higher QPs (even if the difference between the QPs is small) are more likely to be encoded at lower depths (i.e., are less likely to be split), while CUs encoded with lower QPs are more likely to be encoded at higher depths (i.e., are more likely to be split). Therefore, we found that using a different LDF model for each QP improves the classification accuracy of the LDF model. In this paper, we use one model for each different QP used. Therefore, we use three different models for the typical configurations mentioned above.

In addition to this, the proposed transcoder also makes use of a different way of computing the features. The features used here are: (i) the MV Variance Distance (two features); (ii) the MV Phase Variance (two features); (iii) the Number of DCT Coefficients (two features); and (iv) the H.264/AVC Mode Distribution (four features), for a total of ten features.

Before computing the features based on motion vectors for a given region, all the H.264/AVC motion vectors are scaled to the same reference frame (if different reference frames are used). The MVs are then arranged in a list, accounting for the area that each MV represents in the region (i.e., if a given MV is used for a 16×16 region, then 16 copies of that MV are inserted in the list, while if the MV is used for a 4×4 region, the smallest region defined in the H.264/AVC standard, then only one copy of that MV is inserted in the list). If there are intra H.264/AVC MBs within the current region, then no MV is inserted in the list for that H.264/AVC MB. Note that, since the LDF mapping is only used if there is at least one inter H.264/AVC MB in the current CU, then this list is never empty. For instance, for a region of 64×64, there are at most 256 MVs in this list.

The MV Variance Distance is simply defined as $v = \sqrt{\sigma_x^2 + \sigma_y^2}$, where σ_x and σ_y are defined as the variances of each MV component in the list. Similarly, the MV Phase Variance is defined as the variance of the phase (computed as $atan2\left(mv^k.y, mv^k.x\right)$) of each MV in the list.

The number of DCT coefficients feature is defined as the number of nonzero H.264/AVC coefficients encoded in that region, regardless if the H.264/AVC is encoded in intra or inter mode. The H.264/AVC Mode Distribution is simply defined as the area of the region that is encoded with the following modes by the H.264/AVC: (i) SKIP; (ii) 16×16, 16×8 or 8×16; (iii) 8×8, 8×4, 4×8 or 4×4; and (iv) any intra mode.

Some of the features used are computed for the CU using the following method. First, the feature is computed considering the total area of the CU (i.e., for the depth 0, the whole 64×64 region), resulting in a value σ. Then, the feature is computed for all four sub-CUs (i.e., for the four 32×32 regions), resulting in four values $\{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$. Finally, the two features actually used to build the model (and, later, to classify the CU) are defined as: $\{\sigma, max\left(\sigma_0, \sigma_1, \sigma_2, \sigma_3\right)\}$. The rationale to compute the variance for smaller regions is to describe areas that are mostly homogeneous but present a significant difference in a small region. The idea of using just one value (the maximum) for the four smaller regions is that it is not important to describe where this difference happened, it suffices to describe that it does happen. An example of this is given in Fig. 2. This procedure is used for the MV Variance Distance, the MV Phase Variance and the Number of DCT Coefficients.

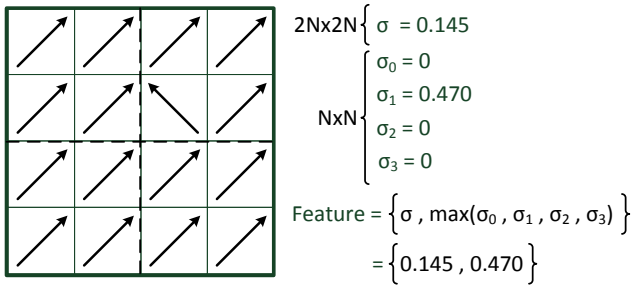

Fig. 2. Computing the MV Phase Variance feature for a CU Region.

IV. Experimental Results

In order to evaluate the proposed transcoder, three transcoding options are compared: (i) the trivial transcoder, using fast ME and fast mode decision (namely, RT-FAST); (ii) the reference transcoder based on content modeling using a dynamic training [8] (namely, RT-LDF); and (iii) the proposed transcoder presented in the previous section. For the H.264/AVC, the reference software JM 14.2 [16] is used, and for the HEVC, the reference software HM 13.1 [17] is used. For all sequences, the QPs are 37, 32, 27 and 22, and the full length of the sequence is transcoded (10 seconds). Both codecs are using a low-delay coding configuration with 1 reference frame. For both RT-LDF and PT, the first 12 inter-frames are used for training. The results are shown using the Bjontegaard Delta bitrate measure [18].

First, we compare the accuracy of the new LDF model compared to the LDF model used in RT-LDF. For this, the first 12 inter-frames of the sequences are used in order to build a model, and then the trivial transcoder is applied to the next 48 frames. The four QPs are used. The LDF classification, using the model obtained with the first 12 frames, is then compared to the result given by the trivial transcoder, which is used as the ground truth. The results are presented in Table II. It can be seen from the table that the proposed method is more accurate than our previous method (RT-LDF) for all but one sequence. However, for this sequence, Tennis, the difference is rather small (0.2%) and it can be explained by the presence of a large amount of intra MBs in the H.264/AVC bitstream. Recall from Sec. III-A that, in RT-LDF, CUs containing any intra MB are not considered for LDF classification, whereas in the new method the LDF classification is only skipped if all H.264/AVC CUs are encoded as intra.

Table I shows the rate-distortion and speed-up results. As expected, since the proposed transcoder tests even less partitions than RT-LDF, it shows a larger rate-distortion loss. However, it also shows a significant speed-up compared to RT-LDF. On average, the proposed transcoder is 1.65 (with a

TABLE I. TRANSCODER RESULTS COMPARED TO RT-FAST.

		BD-Rate %			Speed
Sequence	Method	Low	High	Average	Up
Kimono1	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	2.45	2.41	2.52	2.26
24 Hz	PT	5.12	4.26	4.70	3.50
Tennis	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	1.04	1.34	1.18	1.36
24 Hz	PT	16.5	8.34	12.2	2.32
ParkScene	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	4.74	2.46	3.62	2.62
24 Hz	PT	8.86	3.81	6.27	3.77
Cactus	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	5.90	3.82	4.90	2.37
50 Hz	PT	12.7	6.98	10.1	3.90
BasketballDrive	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	4.58	3.18	3.98	1.80
50 Hz	PT	11.5	5.85	8.77	3.48

TABLE II. RESULTS FOR THE LDF ACCURACY.

Sequence	RT-LDF	PT
Kimono1	74.2%	76.7%
Tennis	77.2%	77.0%
ParkScene	77.3%	80.6%
Cactus	78.9%	79.4%
BasketballDrive	75.8%	77.5%

minimum of 1.44 and maximum 1.93) times faster than RT-LDF. It is important to notice that the coding configuration used in the tests use only one reference frame - higher speed-ups are expected if more reference frames are used. Also, note that the speed-up figures refer to the original HM code - no software acceleration or parallel processing is used. In fact, these techniques could be built on top of the proposed algorithm.

V. CONCLUSION

In this paper we presented a transcoder that performs CU classification based on a machine learning technique with the goal of speeding-up transcoding. The proposed transcoder presents a significant higher speed-up (on average, 1.65 times faster), compared to our previous works, at the cost of a higher loss in bitrate. For future work, we plan to further study the effect of intra macroblocks in the H.264/AVC bitstream (present in sequences such as Tennis) which still presents a challenge to the transcoder. Also, CU classification with more classes (instead of the binary approach used) could be tested in order to further speed-up the transcoder.

REFERENCES

[1] ITU-T, "ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services," ITU-T, Tech. Rep., May 2003.

[2] ——, "H.265: High Efficiency Video Coding." ITU-T, Tech. Rep., June 2013.

[3] ISO/IEC, "Iso/iec 23008-2:2013," ISO/IEC, Tech. Rep., November 2013.

[4] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: An overview of various techniques and research issues," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 793–804, October 2005.

[5] 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, March 2003.

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

[7] T. Shanableh, E. Peixoto, and E. Izquierdo, "MPEG-2 to HEVC video transcoding with content-based modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1191–1196, 2013.

[8] E. Peixoto, B. Macchiavello, E. M. Hung, A. Zaghetto, T. Shanableh, and E. Izquierdo, "An H.264/AVC to HEVC video transcoder based on mode mapping," in *IEEE International Conference on Image Processing (ICIP 2013)*, September 2013, pp. 1972–1976.

[9] D. Zhang, B. Li, J. Xu, and H. Li, "Fast transcoding from H.264 AVC to high efficiency video coding," in *IEEE International Conference on Multimedia and Expo (ICME 2012)*, July 2012, pp. 651 –656.

[10] H. Shen, X. Sun, and F. Wu, "Fast H.264/MPEG-4 AVC transcoding using power-spectrum based rate-distortion optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 746 –755, June 2008.

[11] T. Shen, Y. Lu, Z. Wen, L. Zou, Y. Chen, and J. Wen, "Ultra fast h.264/avc to hevc transcoder," in *Proceedings of the 2013 Data Compression Conference (DCC 2013)*, March 2013, pp. 241–250.

[12] P. Xing, Y. Tian, X. Zhang, Y. Wang, and T. Huang, "A coding unit classification based avc-to-hevc transcoding with background modeling for surveillance videos," in *Proceedings of Visual Communications and Image Processing (VCIP 2013)*, Nov 2013, pp. 1–6.

[13] E. Peixoto and E. Izquierdo, "A complexity-scalable transcoder from H.264/AVC to the new HEVC codec," in *IEEE International Conference on Image Processing (ICIP 2012)*, September 2012, pp. –.

[14] E. Peixoto, T. Shanableh, and E. Izquierdo, "H.264/AVC to HEVC video transcoder based on dynamic thresholding and content modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2013.

[15] T. Shanableh and K. Assaleh, "Feature modeling using polynomial classifiers and stepwise regression," *Neurocomputing*, vol. 73, no. 10-12, pp. 1752–1759, June 2010.

[16] ITU-T, "Joint model (JM), H.264/AVC reference software," Artech House Inc., Tech. Rep. JM 14.2 KTA 1.0, 2008.

[17] K. McCann, B. Bross, W.-J. Han, I. K. Kim, K. Sugimoto, and G. J. Sullivan, "Jctvc-o1002 high efficiency video coding (hevc) test model 13 (hm 13) encoder description," JCT-VC, Tech. Rep., January 2014.

[18] G. Bjontegaard, "Improvements of the BD-PSNR model," ITU-T Telecommunications Standardization Sector, Video Coding Experts Group (VCEG), Tech. Rep. VCEG-AI11, July 2008.