A FAST HEVC TRANSCODER BASED ON CONTENT MODELING AND EARLY TERMINATION

E. Peixoto, B. Macchiavello, E. M. Hung and R. L. de Queiroz

Universidade de Brasilia, Brazil

ABSTRACT

In this paper, a fast transcoding solution from H.264/AVC to HEVC bitstreams is presented. This solution is based on two main modules: a coding unit (CU) classification module that relies on a machine learning technique in order to map H.264/AVC macroblocks into HEVC CUs; and an early termination technique that is based on statistical modeling of the HEVC rate-distortion (RD) cost in order to further speed-up the transcoding. 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 both the CU classification model and the early termination sieves, that are used in the second stage (called the transcoding stage). The solution is tested with well-known video sequences and evaluated in terms of RD and complexity. The proposed method is 3.83 times faster, on average, than the trivial transcoder, and 1.8 times faster than a previous transcoding solution, while yielding a RD loss of 4% compared to this solution.

***Index Terms*—** Transcoding, HEVC, machine learning, early termination.

1. INTRODUCTION

The High Efficiency Video Coding (HEVC), known formally as Recommendation ITU-T H.265 [1] or ISO/IEC 23008-2 [2] and referred here simply as HEVC, was developed by the Joint Collaborative Team on Video Coding (JCT-VC), a collaboration between the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) to replace the current H.264/AVC standard [3]. Converting legacy H.264/AVC bitstreams to the HEVC standard may be done for two main reasons: (i) compatibility; and (ii) storage (or bandwidth) reduction.

The process that converts from one compressed bitstream (called the source or incoming bitstream) to another compressed bitstream (called the transcoded or outgoing bitstream) [4, 5, 6] is called *transcoding*. It is always possible to perform transcoding by fully decoding the source bitstream and completely re-encoding it to the target codec. This process is usually referred as the trivial transcoder. However, different from usual video coding, the transcoder has access to information already encoded in the incoming bitstream, such as the motion information and encoded residuals, which may be used to speed-up the transcoding process, while still producing high-quality decoded video.

We have proposed several different transcoding strategies for the HEVC standard [7, 8, 9, 10]. Our first work is based on simple, fixed thresholds to determine the HEVC coding unit (CU) partitioning based on the H.264/AVC motion vectors (MVs) [7]. Although it presents a good rate-distortion (RD) performance over a range of sequences, its main weakness is the use of fixed thresholds, regardless of the content of the sequences or the conditions of the transcoding (such as the QP). This weakness was tackled by the use of dynamic thresholding [10].

In our most recent works [8, 10], we have proposed a content modeling transcoder using linear discriminant functions (LDFs). Differently from other transcoders based on machine learning found in the literature [11, 12, 13], which are all based on a single, offline training, this transcoder is based on online training, which divides the sequence into two stages: training and transcoding. During the training stage, the trivial transcoder is applied, and some information is gathered both on how the H.264/AVC encoded each region (gathered from the incoming bitstream) and on how the HEVC chose to encode it (gathered from the HEVC decision engine). Then, this information is used to specifically build a model for that sequence and conditions (such as the quantisation parameters, QPs, and coding configuration), that will then be used in the transcoding stage. This transcoder presents a good RD performance, but it offers a limited speed-up, since several modes are still tested for each CUs (and, in the end, only one of these is actually used to encode that CU). Therefore, the transcoder proposed here focus mainly on reducing the transcoder complexity, while attempting to keep the resulting bitrate loss as low as possible.

The main contributions of this paper are: (i) a better CU classification method, that uses a new combination of features computed from the incoming bitstream and builds a separate model for each QP used; (ii) a new early termination strategy, used to speed-up the transcoding; and (iii) a new algorithm

E-mails: eduardopeixoto@ieee.org, {bruno,mintsu}@image.unb.br, queiroz@ieee.org . This work was partially supported by CNPq under grants 476176/2013-1, 500370/2013-3, 151235/2013-9 and 302853/2011-1.

to compute the early termination sieves (or thresholds), that models the distribution of the RD costs for each mode in the HEVC.

2. RELATED WORK

While there is a vast literature in transcoding, there have been relatively few transcoders targeting the HEVC standard. A work based on the power-spectrum rate-distortion optimisation (PS-RDO) method [14] has been proposed to transcode H.264/AVC bitstreams to the HEVC [15]. In this work, the 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). Another approach [16] in speeding-up transcoding to the HEVC focus on implementing an algorithm for multi-core processors, 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 [17] proposes an HEVC transcoder applied to video surveillance. Different transcoding strategies of CU partition termination, PU candidate selection and ME simplification are adopted for different CU categories (background, foreground and hybrid) to reduce the complexity.

On the other hand, there have been several early termination algorithms in the literature, for different video codecs. Naturally, due to the high complexity of the HEVC, it has been tested with many kinds of fast algorithms [18]. Here, we focus on works based on early CU termination. An Early SKIP detection method, which attempts to terminate the CU mode decision if the SKIP mode cost is lower than a threshold, was proposed [19]. Also, a work based on CBF (coding block flag) early termination [20] stops the CU mode decision when the CBF is zero (i.e., when no residual information is left to be encoded). Another work [21] proposes an Early CU Termination to avoid splitting the CU (and thus testing the four sub-CUs) if the best prediction mode for the current CU is found to be the SKIP mode. A different approach [22] for Early CU Termination makes use of Support Vector Machines (SVM) in order to decide for CU termination. The SVM training, however, is made offline.

3. THE PROPOSED TRANSCODER

As in our previous work [8, 10], the proposed transcoder operates in two distinct stages: the *training* and the *transcoding* stages. 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. More training stages, either repetitive (e.g., 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. Therefore, longer sequences could be transcoded dividing it into shorter sequences.

In the training stage, all modes of the HEVC are tested and the H.264/AVC information is used only for training purposes. For each HEVC CU, the following information is gathered and stored: (i) the H.264/AVC features for that CU; (ii) the HEVC chosen mode; (iii) the RD cost for each mode tested; and (iv) the QP used to encode this CU.

The transcoding operations are based on the HEVC CU. The decision always starts at the LCU, used here as 64×64 pixels, and continues recursively to each sub-CUs. Different mapping strategies are used according to the CU depth.

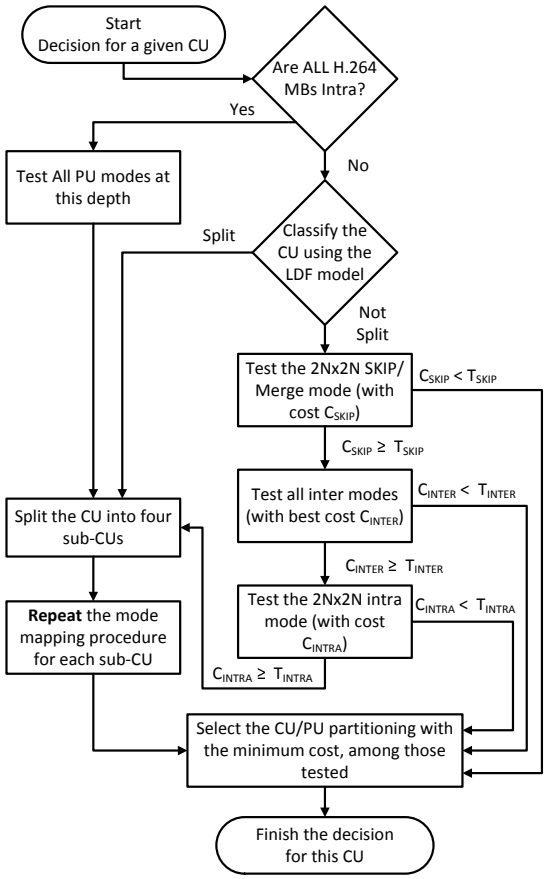

Fig. 1. Algorithm Workflow for depths 0 and 1

For CUs at depths 0 and 1 (64×64 and 32×32 pixels, respectively), the following strategy is used (seen in Fig. 1). For CUs in which all H.264/AVC macroblocks within the CU are encoded in intra mode, all HEVC modes are tested and the CU is split. Otherwise, first the CU is classified in two classes: *split* or *not split*, using a LDF classifier. This classifier uses the H.264/AVC features, and it is fully explained in Sec. 4. For CUs classified as *split*, no partition at this depth is tested - the CU is split into four sub-CUs and the algorithm is applied again for each of these sub-CUs. Otherwise, for CUs classified as *not split*, the SKIP/MERGE mode is tested first, and its RD cost is computed. If this cost is lower than

a sieve (T_{SKIP}, in the figure), then the CU is terminated and it is encoded using the SKIP/MERGE mode. If the cost is higher than T_{SKIP}, then all inter modes are tested (as it is default in the HEVC, only some of the asymmetric partitions are tested). The cost for the best *inter* mode is then compared to another sieve, T_{INTER}. Again, if it is lower than this sieve, the CU is terminated and encoded with the best mode tested so far (either SKIP or INTER - however, only the costs for the inter mode are compared to T_{INTER}, since T_{INTER} is computed using statistics for the inter modes only). If the cost is higher than T_{INTER}, then the intra mode is tested (only the $2N \times 2N$ mode is considered). Again, if the cost of the intra mode is lower than the sieve T_{INTRA}, the CU is terminated and encoded with the best mode tested so far (either SKIP, INTER or INTRA). Otherwise, the CU is split, and the algorithm is applied recursively for each sub-CU.

For CUs at depths 2 and 3 (16×16 and 8×8 pixels, respectively), we have the information on how the H.264/AVC partitioned the region. Thus, a simple mode mapping algorithm is used, in which all partitions larger than the H.264/AVC partitions are tested at this depth. This mapping is described in more details elsewhere [10].

Finally, for all CUs, a simple MV reuse algorithm is used. 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), and the default sub-pixel search is applied at half and quarter-pixel levels.

4. BUILDING THE MODELS

The two most common coding configurations used in the HEVC, namely the low-delay and the random access configurations, make use of a coding structure that varies the base QP used to encode each frame. Usually, for a given base QP used to encode the first intra frame, the subsequent frames are encoded using a QP offset of $\{+3, +2, +3, +1\}$. This radically changes the way the HEVC mode decision works, changing the average RD cost for each CU, the mode distribution, among other changes. Therefore, in this paper we have used a different model for each QP, both for the CU classification and the early termination sieves.

4.1. CU Classification Model

As in our previous works [8, 10], a simple machine learning algorithm is used, the linear discriminant functions [23]. The reason for this choice is that this algorithm shows a good performance with a low complexity training. However, we use different features than our previous works. 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.

Some of the features used, such as the MV Phase Variance, 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: $\{\sigma, max\,(\sigma_0, \sigma_1, \sigma_2, \sigma_3)\}$. 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.

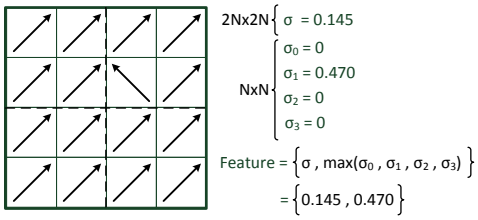

Fig. 2. Computing the MV Phase Variance feature for a CU Region.

The H.264/AVC Mode Distribution is simply defined as the area of the CU 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.

Our tests have shown that using these features, in addition to separating the features by QP, leads to a significant improvement in the classification accuracy of the LDF model. As an example, for ParkScene sequence using QP 37, our previous LDF model [8] achieved 75.6% accuracy in discriminating among the split and not split classes. For the same dataset, the new model achieved 81.13% accuracy (75.6%, 77.7% and 85.56%, for QPs 38, 39 and 40, respectively).

4.2. Early Termination Sieves

During the training stage, the transcoder gathers information about the RD cost of each tested HEVC mode, the chosen mode and the QP used to encode the current CU. This information is used to derive the sieves used in the transcoding stage. As seen in Sec. 3, three sieves are used: T_{SKIP}, T_{INTER} and T_{INTRA}, for the SKIP/MERGE mode, all inter modes (all inter modes are combined producing just one sieve) and the $2N \times 2N$ intra mode (the $N \times N$ and the PCM intra modes are never used). Then, for each of these groups of modes, the following algorithm is applied to derive the sieve.

First, only the minimum cost for the CUs that are actually encoded with that mode are considered. If there are less than ten samples, then the information is regarded as insufficient and no sieve is used (i.e., early termination will never be performed for that particular mode, depth and QP). If more samples are available, they are modeled using a Log-Normal distribution (as seen in Fig. 3). This can be easily done by computing the mean and variance of the samples. The sieve is

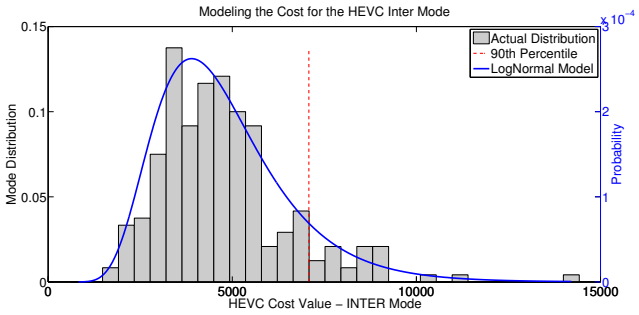

Fig. 3. Example of modeling the HEVC RD Cost as a Log-Normal distribution.

Table 1. Transcoder Results compared to RT-FAST. In order to compute the sieves, the 90-th percentile is used for PT.

		BD-Rate %			Speed
Sequence	Method	Low	High	Average	Up
Kimono1	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	2.66	2.35	2.60	2.29
24 Hz	PT	4.91	4.13	4.54	3.88
Tennis	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	1.15	1.44	1.27	1.37
24 Hz	PT	15.9	9.09	12.3	2.55
ParkScene	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	4.90	2.48	3.69	2.68
24 Hz	PT	8.89	3.92	6.39	4.55
Cactus	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	6.12	3.77	5.00	2.38
50 Hz	PT	10.5	7.23	8.95	4.51
BasketballDrive	RT-FAST	0.00	0.00	0.00	1.00
1920×1080	RT-LDF	4.60	3.22	4.02	1.81
50 Hz	PT	7.87	5.15	6.57	3.70

computed as a percentile using the Log-Normal model built. Note that the sieve is computed using the statistics for that particular mode, which is why only the cost for that mode is compared to the sieve during transcoding.

5. 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 based on LDF and Early Termination (namely, PT). For the H.264/AVC, the reference software JM 14.2 [24] is used, and for the HEVC, the reference software HM 13.1 [25] 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 the RT-LDF, the first 10 inter-frames are used for training, while for the PT the first 12 inter-frames are used for training. More frames are used for the latter to ensure that at least 3 frames are available for each QP.

Table 1 shows the results with PT using the 90-th percentile to compute the sieves. As expected, since the proposed transcoder tests even less partitions than the RT-LDF, it shows a larger RD loss, but it is also significantly faster. On average, the proposed transcoder is 1.83 times faster than RT-LDF. In fact, PT is at least 1.50 times faster than RT-LDF (for Kimono1 sequence at QP 22) and at most 2.29 times faster (for BasketballDrive sequence at QP 37). 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.

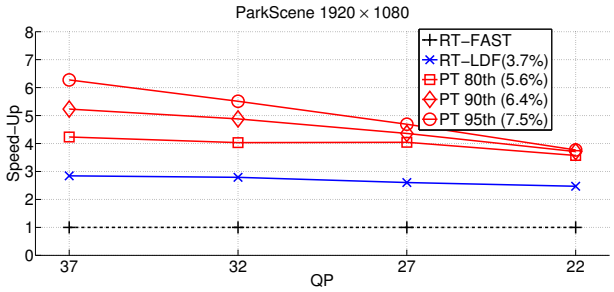

Fig. 4. Speed-up figures for different percentiles. The average bitrate loss is shown in parentheses.

Still, Tennis sequence presents a challenge for transcoding. While PT is still significantly faster than both the trivial transcoder and RT-LDF, it also presents the highest loss. The main challenge of this sequence is due to the high incidence of intra macroblocks in the H.264/AVC bitstream, which impacts the amount of information available.

The effect of varying the percentile is shown in Fig. 4, for ParkScene sequence. As expected, decreasing the percentile (and, therefore, the sieve) yields a lower bitrate loss, but a slower transcoder. However, the bitrate loss is rather small, and the change in speed-up is concentrated in higher QPs. This is expected, since the sieves act first on the SKIP mode, which is used more often for higher QPs.

Finally, for all sequences tested, the PT transcoding loss is perceived as an increase in bitrate, not a decrease in the decoded video quality. For all sequences, the highest PSNR loss is -0.12 dB, for Tennis sequence using QP 37.

6. CONCLUSIONS

In this paper we presented a transcoder that combines CU classification based on a machine learning technique with early termination sieves in order to speed-up transcoding. The proposed transcoder presents a significant higher speed-up (on average, 1.8 times faster), compared to our previous works, at the cost of a slightly higher loss in bitrate (on average, 4%). It also has the ability to trade-off complexity for rate-distortion performance (mostly in the form of bitrate increase), by changing the percentile used to compute the early termination sieves. For future work, we plan to further study the effect of intra macroblocks in the H.264/AVC bitstream, present in the Tennis sequence, which still presents a challenge to the transcoder. Also, we plan to further study the transcoding of the motion information and the transcoding of CUs of lower depths (16×16 and 8×8), as there is still some room for improvement in these areas.

7. REFERENCES

[1] ITU-T. Recommendation H.265: High Efficiency Video Coding., ITU-T, Jun. 2013.

[2] ISO/IEC 23008-2:2013, ISO/IEC, Nov. 2013.

[3] ITU-T, ITU-T Recommendation H.264, Advanced video coding for generic audiovisual services, ITU-T, May 2003.

[4] 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, Oct. 2005.

[5] 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.

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

[7] E. Peixoto and E. Izquierdo. "A Complexity-Scalable Transcoder from H.264/AVC to the new HEVC Codec", in *IEEE Int. Conf. on Image Processing (ICIP 2012)*, pp 737-740, Sep. 2012.

[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 Int. Conf. on Image Processing (ICIP 2013)*, pp. 19721976, Sep. 2013.

[9] E. Peixoto, T. Shanableh and E. Izquierdo, "H.264/AVC to HEVC Video Transcoder based on Dynamic Thresholding and Content Modeling", in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 99-112, Jan. 2014.

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

[11] G. Fernandez-Escribano, P. Cuenca, L. O. Barbosa and H. Kalva, "Very low complexity MPEG-2 to H.264 transcoding using machine learning", in *ACM Int. Conf. on Multimedia (ACM Multimedia 2006)*, ACM, pp. 931940, Oct. 2006.

[12] G. Fernandez-Escribano, H. Kalva, J.L. Martinez, P. Cuenca, L. Orozco-Barbosa and A. Garrido, "An MPEG-2 to H.264 video transcoder in the baseline profile", in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 5, pp. 763-768, May 2010.

[13] C. Holder, T. Pin and H. Kalva, "Improved machine learning techniques for low complexity MPEG-2 to H.264 transcoding using optimized codecs", in *IEEE Int. Conf. on Consumer Electronics (ICCE 2009)*, pp. 1-2, Jan. 2009.

[14] H. Shen, X. Sun and F. Wu, "Fast H.264/MPEG-4 AVC transcoding using powerspectrum based rate-distortion optimization", in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 746 755, Jun. 2008.

[15] D. Zhang, B. Li, J. Xu and H. Li, "Fast transcoding from H.264 AVC to high efficiency video coding", in *IEEE Int. Conf. on Multimedia and Expo (ICME 2012)*, pp. 651656, Jul. 2012.

[16] T. Shen, Yao Lu, Z. Wen, L. Zou, Y. Chen and Jiangtao Wen, "Ultra fast H.264/AVC to HEVC transcoder", in *Proc. of the 2013 Data Compression Conf. (DCC 2013)*, pp. 241-250, Mar. 2013.

[17] 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 *Proc. of Visual Communications and Image Processing (VCIP 2013)*, pp. 1-6, Nov. 2013.

[18] H. Li Tan, F. Liu, Y. H. Tan and C. Yeo, "On fast coding tree block and mode decision for High-Efficiency Video Coding (HEVC)", in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pp. 825828, Mar. 2012.

[19] H.-Mi Yoo and Jae-Won Suh, "Fast Coding Unit decision algorithm based on inter and intra Prediction Unit termination for HEVC", in *IEEE Int. Conf. on Consumer Electronics (ICCE 2013)*, pp. 300301, Jan. 2013.

[20] R. H. Gweon, Y.-L. Lee and J. Lim, "Early termination of CU encoding to reduce HEVC complexity", JCTV -F045, JCT-VC of ITU-T SG16 WP3 and ISO IEC JTC1/SC29/WG11 6th Meeting, Torino, Italy, Apr. 2011.

[21] K. Choi, S.-Hyo Park and E. S. Jang, "Coding tree pruning based CU early termination", JCTV-F092, JCT-VC of ITU-T SG16 WP3 and ISO IEC JTC1/SC29/WG11 6th Meeting, Torino, Italy, Apr. 2011.

[22] X. Shen and Lu Yu, "CU splitting early termination based on weighted SVM", in *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 111, Jan. 2013.

[23] T. Shanableh and K. Assaleh, "Feature modeling using polynomial classifiers and stepwise regression", in *Neurocomputing*, vol. 73, no. 10-12, pp. 17521759, Jun. 2010.

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

[25] 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, Jan. 2014.