FAST PARTITIONING ALGORITHM FOR HEVC INTRA FRAME CODING USING MACHINE LEARNING

Damián Ruiz-Coll[1], *Velibor Adzic*[2], *Gerardo Fernández-Escribano*[1],
Hari Kalva[2], *José Luis Martínez*[1] *and Pedro Cuenca*[1]

[1]Instituto de Investigación en Informática de Albacete (Albacete, SPAIN),
and [2]Florida Atlantic University (Boca Raton, Florida, USA)

ABSTRACT

High Efficiency Video Coding (HEVC) is the new video coding standard recently approved by ISO and ITU. HEVC allows a bit rate reduction greater than a 50% with respect to its predecessor, the H.264/AVC, offering the same perceptual quality, by means of a set of new tools that have been introduced. Compared with the current state of the art in image coding, such as JPEG, JPEG2000 or the new JPEG XR, the new Intra Frame coding performs a high compression process in the "All-Intra" mode. All these improvements are at expense of a high computational cost, making it considerably difficult to implement in real time. Hence, this paper presents a mechanism that can be used by the RDO algorithm to select the optimal coding block size for Intra-Prediction, by using a data mining classifier, based on a previous training. Experimental results show that the proposed algorithm can achieve a 30% of Time Savings over a wide range of high resolution sequences (Class A, B and F), with a negligible loss of coding efficiency.

Index Terms— HEVC, Rate Distortion Optimization, Coding Tree Block, Intra Prediction, Machine Learning.

1. INTRODUCTION

In January 2013 a new milestone in the history of video compression has been achieved with the adoption of the new video coding standard, known as HEVC (High Efficiency Video Coding), developed by the working group JCT -VC (Joint Collaborative Team Video Coding), which is formally standardized as ITU -T H.265 [1] and MPEG-H Part2 (ISO/IEC 23008-2). The new standard was born as a natural evolution of its predecessor the H.264/AVC standard, after a long period of 10-year of undisputed multimedia implementation in all segments of the domestic and professional market. HEVC was initially conceived with a dual purpose: to offer an efficient solution to the strong demand of bandwidth over fixed and wireless networks for mobile multimedia services, and to obtain a high encoding efficiency for formats beyond HD resolution, such as the new 4k and 8k formats, which have been defined in the new standard as Ultra High Definition (UHD).

However, the complexity of the new features included in the new standard is extremely high, such as the intra-frame prediction, making it difficult to implement for real-time applications, especially for high resolution video formats such as HD and UHDTV. This fact motivates the approach presented in this paper, which covers the design of a mechanism to reduce the complexity of the intra-frame partition step within the HEVC coding.

So, the paper is organized as follows. First, in Section 2 we will give a brief introduction of HEVC architecture, focusing on the coding block partitioning, Section 3 reviews literature on complexity reduction in HEVC intra-frame coding optimizations. In Section 4, we will introduce and detail our approach, and show the experimental results in Section 5. The Sections 6 and 7 are for the conclusions and the references respectively.

2. HEVC ARCHITECTURE

HEVC can be considered an evolution of the current H.264/AVC, because it maintains the hybrid coding scheme of H.264/AVC, in which the video stream is temporarily encoded by means of a Motion Estimation and Motion Compensation (ME-MC) process: the residue is compressed in the spatial domain by transform techniques, using an integer version of the Discrete Cosine Transform (DCT). In addition, HEVC introduces small efficiency improvements with respect to H.264/AVC, and incorporates three new features that distinguish it significantly from its predecessors, such as the new coding unit based on a new hierarchical block structure called Coding Tree Unit (CTU), a new Angular Intra Prediction, and a new tool in the decoding loop called Sample Adaptive Offset (SAO), which is applied to the reconstructed samples after the deblocking filter, with the goal of improving the perception in the decoded sequence. These improvements allow us to obtain a globally high level of efficiency with respect to other

---

[1] This work has been jointly supported by the MINECO and European Commission (FEDER funds) under the project TIN2012-38341-C04-04.

standards, but at the cost of a high increase of computational complexity, being necessary to the address algorithms and models with reduces it. In [2] a study of the computational complexity of each of the tools defined in the HEVC is presented, and its impact on the coding scheme. Due to space limitations, a more complete description of each of one of these tools can be found in [3].

2.1. Partitioning of the coding unit (CTU)

HEVC defined the CTU structure as a replacement of the coding unit based on a non-overlapping 16x16 pixels block (macroblock), as was defined in the previous video coding standards. The CTU is a new flexible structure with a maximum size of 64x64 that can be iteratively partitioned into four sub-blocks of half resolution, reaching a minimum allowable size of 8x8 pixels.

As it is described in [4], a CTU is structured in a hierarchical tree where each branch ends in a node that defines the minimum Coding Unit (CU). Each CU is by itself a new root of two new trees, which contain the Prediction Units (PU), and the processing units called Transform Unit (TU). The residue obtained in each of the PUs, is partitioned and transformed using a tree structure; the Residual quadtree (RQT) [5], which allows a maximum of three levels of decomposition. In the Figure 1, an example of the Intra frame partitioning is shown.

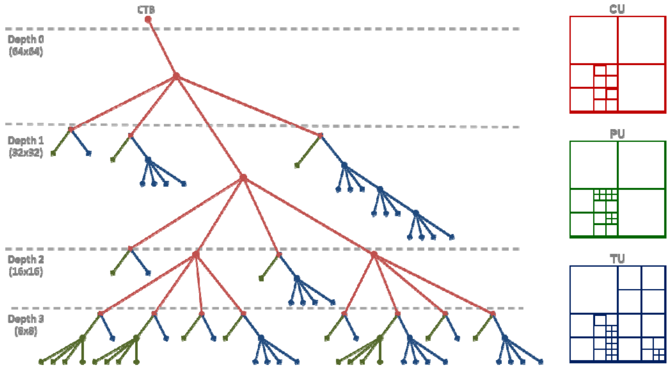

Figure 1. CTU partitioning

As it is expected, this process consumes time, due to the great number of possibilities that have to be evaluated. This is the point where our approach is focused on, as we will detail it in Section 4.

3. RELATED WORK

As with H.264, HEVC uses a well-known Rate Distortion Optimization model (RDO) [6] to achieve the best coding efficiency. RDO reaches the optimal partitioning by evaluating all CU, PU, and TU sizes for each of those CU-PU combinations.

The HEVC intra-frame prediction algorithm defined in HEVC applies the same approach used in H.264/AVC. The high spatial correlation between the pixels of a CU and its neighboring pixels from the top and left blocks is exploited in the construction of the prediction samples. The HEVC Intra-Prediction defines two non-directional modes, DC and Plane mode, and a new "Angular Intra Prediction" [7], increasing the number of directional predictors from 9, used by H.264, to 33.

In order to reduce the high computational complexity of the intra-frame prediction, a number of proposals have been presented. In [8] a fast CU decision using the correlation of the content and the optimal CU depth level is presented. The authors report a 21% time reduction with a rate penalty of 1.7%. Chen et.al [9], show a fast intra algorithm based on pixel gradient statistics and a mode refinement that can achieves a 28% of time saving with coding performance decreasing of 0.5%. In [10] a similar approach based on gradient is shown, achieving almost a 20% time reduction and negligible coding degradation.

A fast intra-frame prediction algorithm using a Rate-Distotortion estimation based on Hadamard Transform is presented [11] reducing the computational complexity by 32%, but with an average performance drop of 1.2%.

4. FAST PARTITIONING ALGORITHM FOR HEVC INTRA-PREDICTION

In order to reduce the brute force approach of traditional RDO, the HM implementation introduce a fast intra-frame prediction algorithm [12][13], which reduces the number of directional predictor candidates, but it still performs an exhaustive CU sizes evaluation, with a high computational cost.

The aim of our approach is to substitute the fast HM implementation used to select the optimal coding block size for intra-frame prediction, by using a data mining classifier, based on a previous training. In [14], a non-traditional use of machine learning in video encoding and transcoding shows that the encoding complexity can be reduced without significant quality losses. An optimal coding block size classifier for intra-frame prediction would then take a 64x64 block and make fast CTU partitioning decisions without exhaustive evaluations.

4.1. Training set and Classifier

With the aim of creating a training set of CB covering a wide range of complexity, we have selected the first frame of a set of 6 high resolution sequences selected from the JCT-VC test sequences [15]. We have chosen two Class A sequences (Traffic and PeopleOnStreat), four Class B sequences (ParkScene and BQTerrace), and two Class F sequences (SliceEdit and BasketballDrillText), which can be consider a sufficiently representative training set because those contain both blocks with homogeneous areas and high-detail textured blocks. Even though the training set is created with the first frame of the video sequences, the spatial and temporal complexity variation in these sequences makes subsequent frames different from the first and hence a good test set. Figure 2 shows the spatio-temporal (ST) information calculated for all test sequences computed according to P.910 recommendation [16]. It is useful to compare the relative spatial information and temporal information found in the test sequences. Generally, the compression difficulty is directly related to the spatial and temporal information of a sequence. For the vertical axis of Figure 2 (Temporal Information), low values correspond to sequences having very limited motion. High values indicate that a sequence contains scenes with a lot of motion. For the horizontal axis (Spatial Information), low values correspond to scenes having minimal spatial detail, and high values are found in scenes having significant spatial detail.

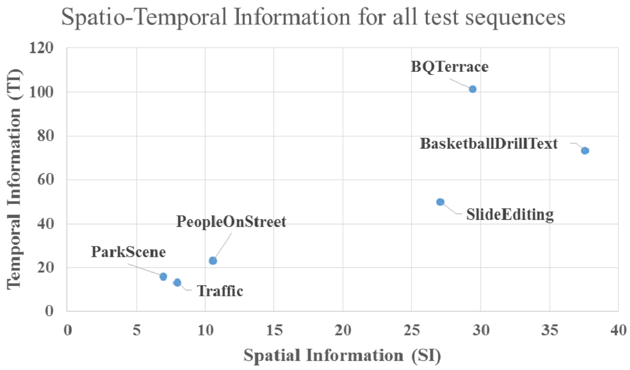

Figure 2. Spatio-Temporal information for the sequences

We use two training set the luma component of the 64x64 CTU and the four sub-CUs of 32x32 of the same CTU. Considering the sequences resolution, our training set is 4231 CUs of 64x64 and 16924 CUs of 32x32. The feature of each of these CU will be an instance for the classifier, which has to be able to figure out the correlation between the CU features and its class. We have used the C.4.5 [17] classifier, because it is shown to performance well for general purpose classifying problems [18], and also for video coding purpose [19].

4.2. Feature selection

Feature selection aims to select a subset of attributes that can lead to a highest performance in classification task of the CBs. There are lots of feature that we can extract from CB to describe its content, such as texture, color or shape, but we need to consider their computational cost because we need a low complexity classifier in order to speed up the partitioning algorithm.

Taking into the count that our approach is to determine when a 2Nx2N block has to be “Split” or “Not-Split”, we have found that comparing the features of the four NxN sub-blocks we can achieve a well accuracy for the CU classification. We have computed a huge number of statistics commonly used in the processing of images applied to CU, such as block entropy, AC energy of DCT coefficients, mean of DC coefficient of DCT, the mean of border error, the block variance, the block mean, the variance of the block rows and columns, between others, and we have trained the classifier with those features.

For the 64x64 CUs, it can be observed that the most relevant attributes for the classifier that appear in the tree are the variance of 64x64 CU, the variance of the variance of the 32x32 sub-blocks, the variance of the means of the 32x32 sub-blocks, that is depicted into the Figure 3. The classifier determines that a CU is homogeneous when the variance between sub-blocks is low, classifying it as 2N, and otherwise if the variance between sub-blocs are large, that means the CU has to be partitioned to be encoded efficiently.

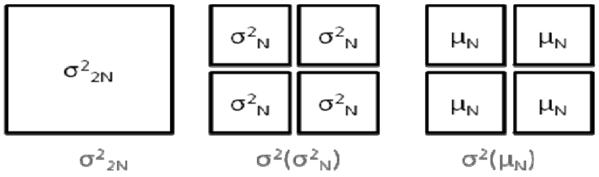

Figure 3. Selected attributes for 64x64 CUs classification

The behavior is very similar for the 32x32 CUs, in this case the variance of 32x32 blocks and the variance of the variance of 16x16 sub-blocks, are showed as the most relevant attributes within tree.

4.3. Decision Tree

The decision tree was made with the Waikato Environment for Knowledge Analysis (WEKA) Version 3.6.10 [20]. WEKA is an effective data mining tool that includes the C.4.5 classifier. The Figure 4 depicts the decision tree build using the training set described before; it is a hierarchical tree with two nodes for 64x64 CUs and 32x32 CUS, named Node64, and 32x32 respectively. If the classifier determines that the 2Nx2N CU has not to be split (left branch), the RDO only for 2N and N sizes are computed, avoiding the evaluation of lower resolution CUs.

The evaluation of the partition N by RDO increases the encoding time, but it avoids misclassification of a CU. This is based on the observation that the 2N blocks and their N split blocks have very similar features and are (is?) difficult to filter by the classifier. Considering that the optimal partitioning obtained for the RDO in the intra-frame prediction has strong dependency with QP value, we have trained the classifier for the four QP values (QP 22, QP27, QP32 and QP37) defined by JCT-VC at [21].

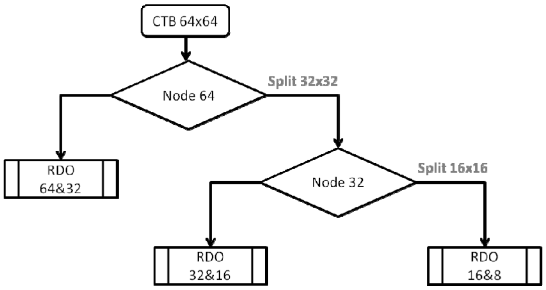

Figure 4. Tree model

4.3. Implementing the classifier

Undoubtedly, the key factor to achieve a fast partitioning algorithm is an efficient implementation of the classifier. We have converted the decision tree rules to if-else sentences in C++, in we have introduced in the HEVC reference software with the follow approach:

1. The attributes are not computed at the beginning of the classifier, instead they are computed when they are required for the inner nodes, so depending of the CU features, some attributes could be not implemented.
2. The variance and mean attributes are computed using integer arithmetic as shown in equations (1) and (2) through bit-shifting instead of division.

$$\mu_N = \left(\sum_{i=0}^{N} x_i\right) \gg log_2 N \quad (1)$$

$$\sigma_N^2 = \left[\sum_{i=0}^{N} x_i^2\right] \gg log_2 N - \left(\sum_{i=0}^{N} x_i \gg log_2 N\right)^2 \quad (2)$$

3. At Node32, the computation of the variance of 32x32 blocks are re-used from the attributes of the Node64 avoiding a new computational cost.

5. SIMULATION RESULTS

The proposed algorithm was implemented on the HEVC reference software HM 10.0 [22], and the experiment was run on Intel® Core™ i7-2600 CPU@3.40GHz platform.

The experiment was conducted under the common test conditions and software reference configurations recommended by the JCT-VC [21] for the "All-Intra" mode and Main profile (AI-Main). We run the experiment for two Class A sequences (Traffic and PeopleOnStreat), two Class B sequences (ParkScene, and BQTerrace), and two Class F sequences (SliceEdit and BasketballDrillText), of the JCT-VC video test sequences [15], encoding the total frames of each sequence. The rate-distortion performance was computed using the BD-rate metric defined by ITU in document [22]. Table I shows the comparison results of our algorithm compared to the HEVC reference implementation.

It can be observed how all the sequences have a very good behavior in term of rate penalty, with the worst case for the Traffic (Class A) and ParkScene (Class B) sequences with a rate penalty of 0.9%. The BQTerrace (Class B) sequence shows the best time reduction, achieving a 30.48%. The global experimental results demonstrate that the proposed algorithm can save a 28% of computational complexity on average with a slightly rate increasing of 0.6%.

Table I. Simulation results for Class A (Traffic, PeopleOnStreet), B (ParkScene, BQTerrace) and F sequences (BasketballDrillText, SlideEditing)

Sequence	Resolution	Frames	T. Saving (%)	BD-Rate
Traffic	2560x1600	150	28.81	0.9
PeopleOnStreet	2560x1600	150	25.83	0.5
ParkScene	1920x1080	240	29.70	0.9
BQTerrace	1920x1080	600	30.48	0.7
BasketballDrillText	832x480	500	26.93	0.4
SlideEditing	1280x720	300	26.57	0.3
Claas A (8-bits)			27.34	0.7
Class B			30.09	0.8
Class C			26.75	0.3
Total (A+B+C)			**28.07**	**0.6**

The Figure 5 depicts the rate-distortion curves of HEVC reference software (HM10.0) and the proposed algorithm for the simulation sequences in Table I, showing the negligible differences in terms of quality.

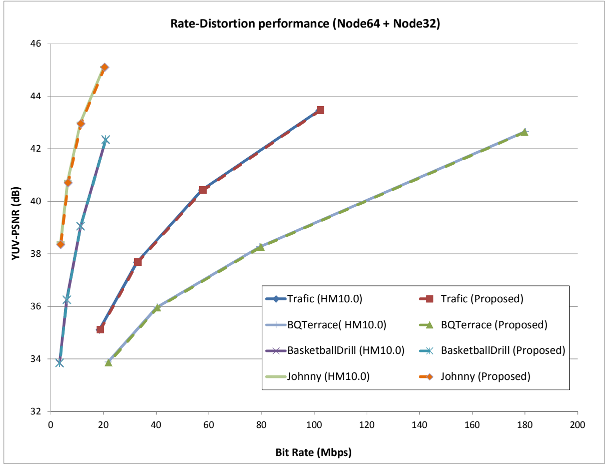

Figure 5. RD curves

6. CONCLUSIONS

We have presented a new approach for accelerating the CU partitioning decision of the HEVC Intra-Prediction, based on trained data mining trees. We have implemented a decision tree using a low complexity attributes, such as mean and variance of blocks, which allows an early classification of the CU avoiding the RDO evaluation of the whole sizes. As has been presented, our approach can reduce a 28% of computational complexity on average, with a negligible rate increasing less than a 0.6% in term of BD-rate (piecewise cubic). Our algorithm achieves good savings even for the sequences with high temporal information, which indicates that it doesn't depend on similarity between training frame and the rest of the frames.

7. REFERENCES

[1] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, Y.-K. Wang, T. Wiegand, “High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent)”, Doc. JCTVC-L1003_v10, Jan. 2013.

[2] Correa, G.; Assuncao, P.; Agostini, L.; da Silva Cruz, L.A., "Performance and Computational Complexity Assessment of High-Efficiency Video Encoders," Circuits and Systems for Video Technology, IEEE Transactions on, vol.22, no.12, pp.1899,1909, Dec. 2012.

[3] Sullivan, G.J.; Ohm, J.; Woo-Jin Han; Wiegand, T.; Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) Standard," Circuits and Systems for Video Technology, IEEE Transactions on, vol.22, no.12, pp.1649,1668, Dec. 2012.

[4] Han, W.-J.; Min, J.; Kim, I.-K.; Alshina, E.; Alshin, A.; Lee, T.; Chen, J.; Seregin, V.; Lee, S.; Hong, Y. M.; Cheon, M.-S.; Shlyakhov, N.; McCann, K.; Davies, T.; Park, J.-H.,"Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools," IEEE Transactions on, vol.20, no.12, pp.1709-1720, Dec. 2010.

[5] K. Panusopone, X. Fang, L. Wang, “Efficient Transform Unit Representation,” Doc. JCTVC-D250, Jan. 2011.

[6] L. Xiang, M. Wien, and J. R. Ohm, “Rate-complexity-distortion optimization for hybrid video coding,” Circuits and Systems for Video Technology, IEEE Trans on, vol. 21, no. 7, pp. 957–970, Jul. 2011.

[7] J. Min, S. Lee, I. Kim, W.-J. Han, J. Lainema, and K. Ugur, Unification of the Directional Intra Prediction Methods in TMuC, JCTVC-B100, Geneva, Switzerland, Jul. 2010.

[8] Liquan Shen; Zhaoyang Zhang; Ping An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," Consumer Electronics, IEEE Transactions on, vol.59, no.1, pp.207-213, Feb. 2013.

[9] Gaoxing Chen; Zhenyu Pei; Lei Sun; Zhenyu Liu; Ikenaga, T., "Fast intra prediction for HEVC based on pixel gradient statistics and mode refinement," Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on, pp. 514-517, 6-10 Jul. 2013.

[10] Wei Jiang; Ma, Hanjie; Yaowu Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, pp.1836-1840, 21-23 Apr. 2012.

[11] Younhee Kim, DongSan Jun, Soon-heung Jung, Jin Soo Choi, and Jinwoong Kim, "A Fast Intra-Prediction Method in HEVC Using Rate-Distortion Estimation Based on Hadamard Transform," ETRI Journal, vol. 35, no. 2, pp. 270-280, Apr. 2013.

[12] Y. Piao, J. H. Min, and J. Chen, Encoder Improvement of Unified Intra Prediction, JCTVC-C207, JCT-VC of ISO/IEC and ITU-T, Guangzhou, Chnia, Oct. 2010.

[13] L. Zhao, L. Zhang, S. Ma, and D. Zhao, “Fast mode decision algorithm for intra prediction in HEVC,” IEEE Visual Commun. Image Process., pp. 1–4, Nov. 2011.

[14] Gerardo Fernández-Escribano, Rashid Jillani, Christopher Holder, Hari Kalva, Jose Luis Martinez Martinez, and Pedro Cuenca, “Video encoding and transcoding using machine learning,” Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008 (MDM '08), 9th International Workshop on, pp. 53-62, Las Vegas, Nevada, 24-27 Aug. 2008.

[15] JCT-VC test sequences. [Online] ftp://hevc@ftp.tnt.uni-hannover.de/testsequences/

[16] Recommendation ITU-T P.910, “Subjective Video Quality Assessment Methods for Multimedia Applications,” International Telecommunication Union, Geneva (1999).

[17] Quinlan, J.-R., “C4.5: Programs for machine learning,” Morgan- Kaufmann, San Francisco, 1993.

[18] Long-Sheng Chen; Jui-Yu Lin, "A study on review manipulation classification using decision tree," Service Systems and Service Management (ICSSSM), 2013 10th International Conference on , vol., no., pp.680,685, 17-19 July 2013.

[19] Gerardo Fernández-Escribano, Hari Kalva, Pedro Cuenca, Luis Orozco-Barbosa, Antonio Garrido, “A Fast MB Mode Decision Algorithm for MPEG-2 to H.264 P-Frame Transcoding,” Circuits and Systems for Video Technology, IEEE Trans on, vol. 18, no. 2, pp. 172–185, Feb. 2008.

[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009). The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[21] F. Bossen, Common Test Conditions and Software Reference Configurations, document JCTVC-L1100, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), 12th Meeting: Geneve, CH 14 – 23 Jan. 2013.

[22] Joint Collaborative Team on Video Coding Reference Software, ver. HM 10.0 [Online]. Available: https://hevc.hhi.fraunhofer.de/.

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves", ITU-T SG16 Q.6 Document, VCEG-M33, Austin, US, Apr. 2001.