A METHOD FOR EARLY-SPLITTING OF HEVC INTER BLOCKS BASED ON DECISION TREES

Guilherme Correa[1], Pedro Assuncao[2], Luciano Agostini[3], Luis A. da Silva Cruz[1]

[1] Instituto de Telecomunicações – University of Coimbra – Coimbra, Portugal
[2] Instituto de Telecomunicações – Polytechnic Institute of Leiria – Leiria, Portugal
[3] GACI – CDTEC – Federal University of Pelotas – Pelotas, Brazil
{guilherme.correa@co.it.pt, amado@co.it.pt, agostini@inf.ufpel.edu.br, luis.cruz@co.it.pt}

ABSTRACT

The High Efficiency Video Coding (HEVC) standard provides a large improvement in terms of compression efficiency in comparison to its predecessors, mainly due to the introduction of new coding tools and more flexible data structures. However, since much more options are tested in a Rate-Distortion (R-D) optimization scheme, such improvement is accompanied by a significant increase in the encoding computational complexity. We propose in this paper a novel method for efficient early-splitting decision of inter-predicted Coding Blocks (CB). The method employs a set of decision trees which are trained using information from unconstrained HEVC encoding runs. The resulting early-splitting decision process has an accuracy of 86% with a negligible computational overhead and an average computational complexity decrease of 42% at the cost of a very small Bjontegaard Delta (BD)-rate increase (0.3%).

Index Terms— inter mode decision, early-splitting, data mining, decision trees, HEVC

1. INTRODUCTION

The High Efficiency Video Coding (HEVC) standard has achieved significant compression efficiency in comparison to its predecessors by using several new coding tools and much more flexible structures, such as the quadtree-based Coding Blocks (CB), the tree-based Prediction Blocks (PB) and the quadtree-based Transform Blocks (TB). However, to achieve the best Rate-Distortion (R-D) performance, the current reference implementation of the HEVC standard employs an exhaustive process which tests every possible combination of encoding structures and chooses the one with lowest R-D cost. This process, though yielding optimal encoding efficiency, greatly increases the encoder's computational complexity in comparison with its predecessors, limiting its use in computationally and energy constrained environments.

According to [2], the operations involved in inter-frame prediction correspond to about 60% of the encoding computational complexity of HEVC. For each CB, the encoder initially tests the *Merge* mode, which is conceptually similar to the *SKIP* mode of H.264/AVC. The *Merge* mode, also known as *Merge/SKIP Mode* (MSM) allows deriving motion information from spatially or temporally neighboring PBs, forming a merged region sharing the same motion information. After testing MSM, all the remaining partitioning modes are tested in the order presented in Fig. 1, except in 8×8 CBs, which allow only the first four partitioning modes (MSM, *2N×2N*, *2N×N* and *N×2N*). For each PB in each mode, the encoder tests all candidate motion vectors and chooses the best option in terms of R-D cost. However, as only the best motion vectors and the best inter modes are chosen, most of the computation performed in this Rate-Distortion Optimization (RDO) process is discarded, summing up to large amounts of wasted encoding time. Consequently, fast mode decision (FMD) methods that simplify the decision procedure allowing lower-complexity implementations of HEVC encoders are potentially very useful.

Several methods for reducing or dynamically controlling the computational complexity of HEVC focusing on low-complexity encoding structure definition and FMD approaches have been already been published in the literature. In [3-5], spatial and/or temporal correlations among CBs are used to decrease the complexity involved in the coding tree structure definition. In [6, 7], temporal correlation among CBs is also used to decide which inter modes should be tested. In [7], temporal, spatial and tree depth correlation are used to simplify the inter mode decision.

Even though all these works are able to reduce computational complexity to a certain extent, they come with associated losses in terms of R-D efficiency, which are mostly non-negligible. In order to reduce R-D efficiency losses, intelligent approaches which apply machine learning techniques to benefit from intermediate encoding results and image characteristics have been proposed by some authors, especially for transcoding [8, 9] and encoding efficiency optimization [10]. However, no work has been proposed yet which makes use of this type of technique for reducing the computational complexity of the HEVC encoding process.

[1] This work was supported by IT-Portugal, CNPq-Brazil, and the project FCT-CAPES (4.4.1.00 CAPES) (FCT/1909/27/2/2014/S) by FCT-Portugal and CAPES-Brazil.

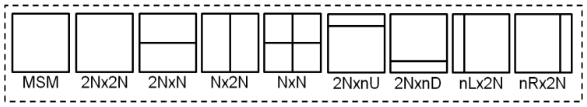

Fig. 1. Inter prediction modes available in HEVC.

In this paper, we present an approach which uses data mining as a tool to build a set of decision trees that are able to predict whether each CB should be split into smaller PBs for further inter prediction. The rest of this paper is organized as follows. Motivations and statistical analysis are presented in section 2. The approach proposed in this work is detailed in section 3. Experimental results are presented and commented in section 4. Finally, conclusions are drawn in section 5.

2. MOTIVATION AND STATISTICAL ANALYSIS

This section presents a statistical analysis performed to unveil characteristics of an HEVC signal that can be useful for training the decision trees central to the algorithm. As previously mentioned, the HEVC standard provides a highly flexible encoding structure. However, even though the HM encoder tests every possibility in the RDO process, the occurrence of each inter partition mode is not equally likely and some of them are used only sporadically, as shown in Fig. 2. The results presented in Fig. 2 were collected for 10 video sequences of different temporal and spatial resolution, motion activity and texture (*BlowingBubbles*, *RaceHorses*, *PartyScene*, *BQMall*, *SlideShow*, *vidyo1*, *ParkScene*, *BasketballDrive*, *NebutaFestival*, *Traffic*) encoded with Quantization Parameter (QP) 32 and using the Random Access (RA) encoder configuration. For other QP values the observed mode frequencies are very similar.

Clearly, most CBs are encoded as only one PB (as MSM or *2N×2N*). For example, almost 95% and 82% of 8×8 and 16×16 CBs are encoded as MSM, respectively. Even though the remaining modes are rarely used, they are still tested for every CB, which is not profitable when trading off between encoding efficiency and computational complexity. However, as the experiments in section 4 will show, simply removing indiscriminately the possibility of using PBs smaller than *2N×2N* for all CBs is not a good solution, since it causes large drops in the encoder R-D efficiency. If there was a way of predicting whether a CB should be split into PBs smaller than *2N×2N*, the cost of testing the remaining inter modes could be avoided in 94% of the CBs, greatly reducing the encoding computational complexity.

In order to find features which could lead to good splitting decisions, we have collected a large amount of data from both the original video sequence and internal encoding variables. More than 40 attributes have been collected per CB and their analysis showed that the MSM R-D cost, the *2N×2N* mode R-D cost, the splitting decision in the CB at the previous tree depth and the *2N×2N* mode residue variance are those which presented higher correlation with the best splitting decision.

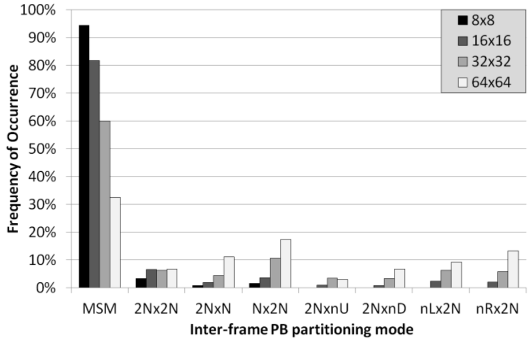

Fig. 2. Probability of occurrence of each inter prediction mode for different CB sizes (QP 32, RA configuration).

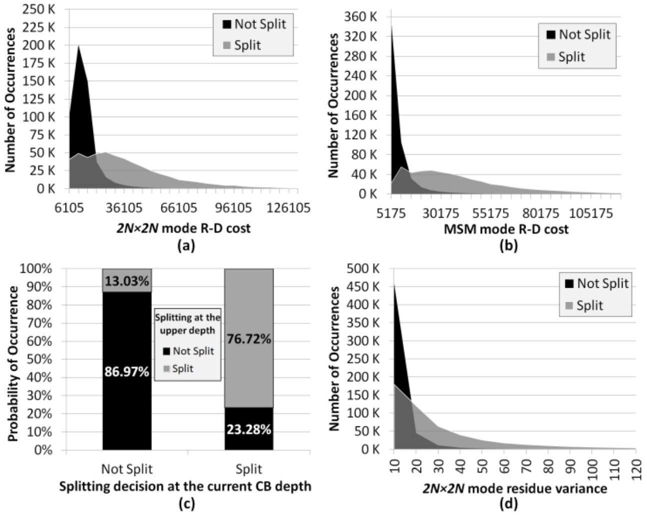

Fig. 3. Analyzed attributes in 16×16 CBs: (a) R-D cost for the *2N×2N* mode, (b) R-D cost for MSM, (c) correlation with splitting decision in upper depth, (d) residue variance for the *2N×2N* mode.

Fig. 3 presents, as an example, statistical results for inter coded 16×16 CBs in the *BasketballDrive* sequence coded with QP 32. For a fair analysis we have randomly selected a data set with 50% of CBs which have been split into smaller PBs and 50% of CBs which have not been split into smaller PBs after the RDO process. It is possible to notice in Fig. 3(a) and Fig. 3(b) that the R-D costs of *2N×2N* and MSM modes are generally lower in those CBs that do not split into smaller PBs than in CBs which split. Fig. 3(c) shows that the splitting decision in CBs at the previous coding tree depth is correlated with the splitting decision at the current tree depth. In 76.72% of the cases when a 16×16 CB was split into smaller PBs, its upper CB was also split into smaller PBs. Analogously, in 86.97% of the cases when a 16×16 CB was not split into smaller PBs, its corresponding upper depth CB was also not split. The distribution of variances in the *2N×2N* mode prediction residues is shown in Fig. 3(d) and also reveals correlation with the splitting decision.

Even though Fig. 3 presents results for one case, all video sequences and the remaining CB sizes (8×8, 32×32 and 64×64) presented similar characteristics, though with different number of occurrences and ranges for the R-D costs and residue variances. We point out that results similar to those of Fig. 3(c) cannot be collected for 64×64 CBs, since those CBs are located at the root of the coding tree and so do not have upper CBs.

The statistical analysis presented in this section provided insights that lead to the approach proposed in this paper. It is clear that the residue information of *2N×2N* PBs, the R-D costs from MSM and *2N×2N* modes and the splitting information from the upper CB can be used to determine the necessity and advantage of splitting the current CB into smaller PBs. The next section presents the approach used to determine the early-splitting conditions for the inter mode decision using such attributes.

3. CODING BLOCK SPLITTING DECISION TREES

Predictive data mining techniques aim at determining the value of a dependent variable by looking at the value of some attributes. There are several methods of predictive data mining currently available, which vary broadly from one another in terms of efficiency, complexity and applicability. Decision trees [11] are commonly used due to their low-complexity implementation, in which a dependent variable can assume one among a finite number of values. In the proposed approach, the dependent variable is, therefore, binary and can assume one of two possibilities: splitting or not splitting the current CB into smaller PBs.

For data mining the four attributes analyzed in Fig. 3 and to assist the development of the early-splitting conditions, we have used the Waikato Environment for Knowledge Analysis (WEKA), version 3.6 [12]. The HM software, version 12.0 (HM12) [1], was set to encode the 10 video sequences mentioned in section 2 with QPs 22, 27, 32, 37, 42, and 47, using the RA configuration.

3.1. Attribute Optimization

Even though the four selected attributes presented the same behavior for all videos and QPs as that shown in Fig. 3, the ranges of R-D costs and residue variance change from one sequence to another depending on specific characteristics, such as spatial and temporal resolution, texture, motion activity, bit depth, and frame rate. Besides, both variance and R-D cost values are dependent on the QP value. Therefore, to allow the use of all video sequences and QPs in the training of the decision trees, a normalization was applied to all attribute samples collected. For each sample, the R-D costs of *2N×2N* and MSM modes and the residue variances of the *2N×2N* PB prediction were divided by the central tendency (mean) of that attribute in the previous frame.

Besides the normalization, an exploration on the relationship between the attributes' values and the splitting decision took place in order to detect if a deeper knowledge could be drawn from the obtained data. This step showed that the ratio between the R-D costs of *2N×2N* and MSM modes yields a better correlation with the splitting condition than the R-D costs alone. It was observed that the number of split cases is much higher for small ratios, which is understandable: a MSM R-D cost higher than a *2N×2N* R-D cost (small ratio) means that performing ME/MC even in a large PB (*2N×2N*) is more advantageous than not performing it at all (MSM), and so further processing might be beneficial.

Fig. 4(a) and Fig. 4(b) present the distribution of ratios before and after the normalization, respectively, for the same case of Fig. 3. By comparing Fig. 4(a) and Fig. 4(b) to Fig. 3(a) and Fig. 3(b) it is possible to perceive that the ratio between the two R-D costs is more correlated to the splitting than the costs taken individually.

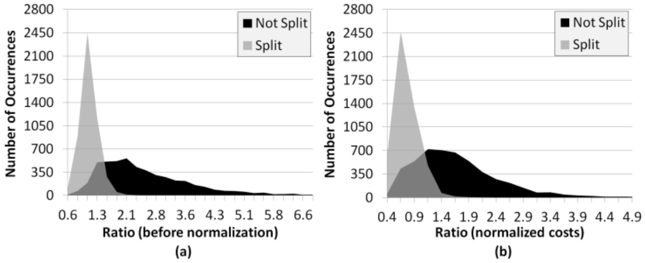

Fig. 4. Ratio between *2N×2N* and MSM R-D costs (a) before and (b) after normalizing the values.

3.2. Growing Decision Trees

WEKA uses ARFF (Attribute-Relation File Format) files as input, which are text files describing a list of instances sharing a set of attributes. Four decision trees, one for each CB size, were grown from ARFF files containing all the attributes to be used in the training step. As shown in Fig. 2, most CBs are not split into smaller PBs, leading to a class imbalance situation that hinders the classifier performance. In order to solve this problem, following common practice [13] we have re-sampled the number of instances corresponding to non-split CBs, so that 50% of the instances in the ARFF files correspond to split CBs and 50% to non-split CBs.

The final attributes used in the training process were, therefore:

- Absolute *2N×2N* R-D cost (`abs_2Nx2N`),
- Absolute MSM R-D cost (`abs_MSM`),
- Normalized *2N×2N* R-D cost (`nor_2Nx2N`),
- Normalized MSM R-D cost (`nor_MSM`),
- Ratio between `abs_2Nx2N` and `abs_MSM` (`abs_ratio`),
- Ratio between `nor_2Nx2N` and `nor_MSM` (`nor_ratio`),
- Absolute *2N×2N* residue variance (`abs_2Nx2N_var`),
- Normalized *2N×2N* residue variance (`nor_2Nx2N_var`),
- Splitting decision at the upper CB depth (`Usplit`).

For each inter coded CB, an instance was created in the ARFF files containing the 9 attributes listed above. Besides the attributes described, the splitting decision is also part of the training vector built from each training CB data. The J48 method, which is an implementation of the C4.5 algorithm [11], was used to create the decision trees.

Fig. 5 shows one of the decision trees, namely the one for 16×16 CBs, which is composed of only 5 decision levels, 9 nodes and 10 leaves. In Fig. 5, leaves “1” and “0” correspond to the *split* and *not split* decisions, respectively, and labels “Y” and “N” correspond to true and false results for the test in the node above, respectively. The other decision trees have similar structures, with a maximum number of decision levels equal to 7 for the case of 32×32 CBs.

4. EXPERIMENTAL RESULTS

In order to evaluate the method proposed in this paper, three encoder versions were compared: the original HM12, the modified HM12 with only MSM and *2N×2N* modes enabled, and the low-complexity encoder proposed. These versions are from now on referred as *original*, *simple* and *proposed*, respectively. All tests were performed using QPs 27, 32, 37, and 42, the RA configuration and 10 video sequences (*PeopleOnStreet*, *SteamLocomotive*, *Kimono1*, *Cactus*, *BQTerrace*, *BasketballDrill*, *BQSquare*, *BasketballPass*, *ChinaSpeed*, *SlideEditing*), none of which was used when training the decision trees.

Initially, the performance of the decision trees was evaluated by counting the number of correct decisions with reference to the decisions taken by the original encoder. On average, the trees performed a correct splitting decision in 86% of the CBs. The decision accuracy was evaluated for each tree separately and the average results are presented in Table 1. In 89.7% of the cases in which the CB was split in the original encoder, the proposed encoder also decided correctly for splitting it. Analogously, in 84.4% of the cases in which the CB did not split in the original encoder, the proposed encoder did not split it. Regarding the wrong decisions, it is important to notice that R-D efficiency losses would occur only in the first case listed (10.3%), when a CB should be split into smaller PBs but is not. This is because in such case a non-optimal inter prediction mode is chosen for the CB. In the second wrong decision case listed (15.6%), the encoder still chooses an optimal mode through RDO, since all inter modes are tested. When comparing the inter modes chosen for each CB in the original and the proposed encoder, it was observed that the latter performed a correct inter mode decision in 97.6% of the cases.

The R-D efficiency of the proposed method was evaluated by comparing the bit rate and PSNR differences between the original encoder and both the simple and proposed versions. R-D efficiency results are presented in Fig. 6 and in Table 2. Fig. 6 presents results for two video sequences, but the remaining cases presented similar behavior. It is possible to perceive that the R-D efficiency achieved with the proposed method is much closer to the original encoder than that obtained by the simple encoder. In fact, even with a 400% zoom applied to the chart, the curves corresponding to the original and proposed encoders are overlapped.

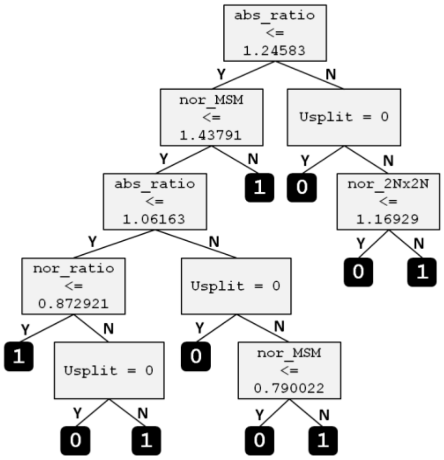

Fig. 5. Decision tree for the PB splitting decision in 16×16 CBs.

		Proposed	
		Split	Not Split
Original	Split	89.7 %	10.3 %
	Not Split	15.6 %	84.4 %

Table 1. Average decision tree accuracy.

Table 2 presents results for the 10 sequences encoded with the simple and the proposed encoders. ΔT indicates the percentage of computational complexity savings and BD-BR indicates the Bjontegaard-Delta (BD)-rate [14] percentage increase when each encoder is used in comparison to the original one. An average BD-BR increase of 0.3% and an encoding time reduction of 42% are observed when the proposed encoder is used, whereas the simple encoder yields a 2.8% BD-BR increase and a ΔT of 60%. The BD-BR/ΔT ratio was calculated to compare the two encoders in terms of BD-BR increase per unit of computational complexity saved. The average BD-BR/ΔT ratio of the proposed encoder is 6.9 times smaller than the average BD-BR/ΔT ratio of the simple encoder, which means that the proposed method performs a much more efficient complexity reduction. Similar results for BD-PSNR (BD-PSNR/ΔT ratio) were presented in Fig. 7 for each sequence. The average BD-PSNR/ΔT ratio of the proposed encoder is 0.02, which is 12 times smaller than the observed in the simple encoder (0.24).

We have compared our results to previous works on complexity reduction of HEVC encoding that presented BD-rate and/or BD-PSNR results using the original HM encoder as reference [3, 5, 15, 16]. Among these works, [15] is the one which presented the largest time savings (41.4%), which is similar to that obtained in our proposal. However, the method in [15] results in a BD-BR/ΔT ratio of 4.54, which is much higher than the cost of our method. We have computed the BD-BR/ΔT ratios for all the comparable works [3, 5, 15, 16], which are 3.20, 1.04, 1.23, and 3.85, respectively, all of which are higher than the 0.71 BD-BR/ΔT ratio of our method.

Video*	Simple			Proposed		
	ΔT (%)	BD-BR (%)	BD-BR/ΔT	ΔT (%)	BD-BR (%)	BD-BR/ΔT
BQSquare	-60	+2.7	4.55	-41	+0.1	0.26
BQTerrace	-61	+1.0	1.67	-47	+0.1	0.14
BasketballDrill	-60	+2.4	4.05	-37	+0.2	0.64
BasketballPass	-60	+3.9	6.43	-37	+0.1	0.21
Cactus	-60	+2.4	4.01	-35	+0.3	0.64
ChinaSpeed	-59	+5.5	9.32	-43	+0.3	0.77
Kimono1	-61	+1.9	3.06	-38	+0.6	1.36
PeopleOnStreet	-61	+3.6	5.93	-42	+0.4	1.51
SlideEditing	-61	+2.0	3.34	-29	-0.1	0.14
SteamLocomotiveTrain	-61	+4.0	6.65	-55	+1.0	2.00
Average	**-60**	**+2.8**	**4.68**	**-42**	**+0.3**	**0.71**

* Sequences **not** used in the training of the decision trees.

Table 2. Complexity reduction and R-D efficiency.

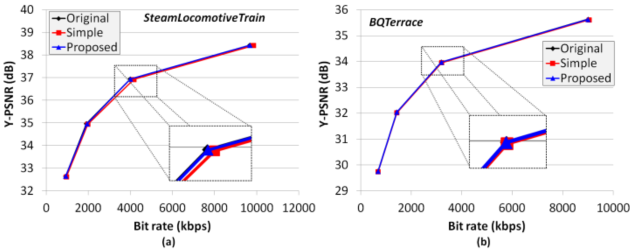

Fig. 6. Rate-Distortion efficiency of the proposed method for the (a) *SteamLocomotiveTrain* and (b) *BQTerrace* video sequences.

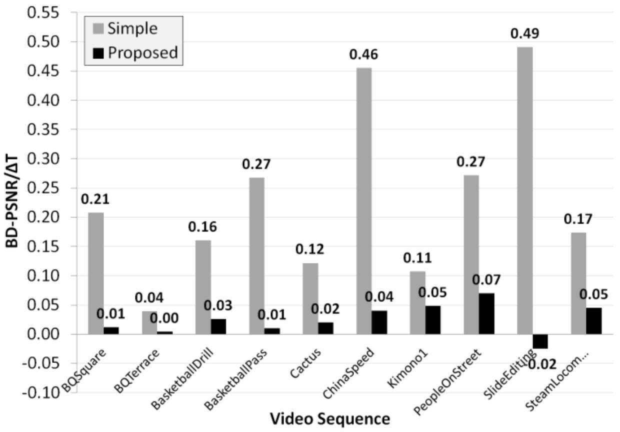

Fig. 7. BD-PSNR/ΔT ratio for each video sequence.

5. CONCLUSIONS

An early-splitting decision method for the HEVC inter prediction is presented in this paper making use of data mining tools for the construction of decision trees. A statistical analysis was performed on a set of attributes to assess their usefulness in the training of four decision trees to be used for deciding whether a CB must be split into smaller PBs during the inter prediction procedure of HEVC. The obtained trees yielded an accuracy of 86% in the splitting decision and resulted in 97.6% of inter modes being correctly chosen. The use of the method in the HM encoder allowed a computational complexity reduction of 42% at the cost of negligible increases in R-D efficiency results (an average increase of 0.3% in terms of BD-rate).

REFERENCES

[1] K. McCann, *et al.*, "High Efficiency Video Coding (HEVC) Test Model 12 (HM 12) Encoder Description," Document JCTVC-N1002, JCT-VC Metting: ed. Vienna, Austria, 2013.

[2] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 22, pp. 1685-1696, 2012.

[3] C. Zhou, F. Zhou, and Y. Chen, "Spatio-temporal correlation-based fast coding unit depth decision for high efficiency video coding," *Journal of Electronic Imaging,* vol. 22, pp. 043001-043001, 2013.

[4] S. Liquan, Z. Zhaoyang, and A. Ping, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. on Consumer Electronics,* vol. 59, pp. 207-213, 2013.

[5] S. Liquan, *et al.*, "An Effective CU Size Decision Method for HEVC Encoders," *Multimedia, IEEE Transactions on,* vol. 15, pp. 465-470, 2013.

[6] G.-Y. Zhong, X.-H. He, L.-B. Qing, and Y. Li, "Fast inter-mode decision algorithm for high-efficiency video coding based on similarity of coding unit segmentation and partition mode between two temporally adjacent frames," *Journal of Electronic Imaging,* vol. 22, pp. 023025-023025, 2013.

[7] J.-H. Lee*, et al.*, "Novel Fast PU Decision Algorithm for the HEVC Video Standard," in *IEEE International Conference on Image Processing*, Melbourne, Australia, 2013.

[8] G. F. Escribano, P. Cuenca, L. O. Barbosa, and H. Kalva, "Very low complexity MPEG-2 to H.264 Transcoding Using Machine Learning," in *Proceedings of the 14th Annual ACM International Conference on Multimedia*, Santa Barbara, CA, USA, 2006, pp. 931-940.

[9] G. Fernandez-Escribano*, et al.*, "Low-Complexity Heterogeneous Video Transcoding Using Data Mining," *Trans. Multi.,* vol. 10, pp. 286-299, 2008.

[10] R. Garcia, D. R. Coll, H. Kalva, and G. Fernandez-Escribano, "HEVC Decision Optimization for Low Bandwidth in Video Conferencing Applications in Mobile Environments," in *IEEE International Conference on Multimedia and Expo*, San Jose, USA, 2013.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*: Morgan Kaufmann Publishers, 1993.

[12] M. Hall*, et al.*, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.,* vol. 11, pp. 10-18, 2009.

[13] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," in *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI2000)*, 2000, pp. 111-117.

[14] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Document VCEG-M33, VCEG Meeting: ed. Austin, USA, 2001.

[15] S. Xiaolin, Y. Lu, and C. Jie, "Fast coding unit size selection for HEVC based on Bayesian decision rule," in *2012 Picture Coding Symposium*, 2012, pp. 453-456.

[16] K. Jaehwan, Y. Jungyoup, W. Kwanghyun, J. Byeungwoo, "Early determination of mode decision for HEVC," in *2012 Picture Coding Symposium*, 2012, pp. 449-452.