Jiménez-Moreno, Amaya, Martínez-Enríquez, Eduardo, Díaz-de-María, Fernando. (2017). Bayesian adaptive algorithm for fast coding unit decision in the High Efficiency Video Coding (HEVC) standard. *Signal Processing: Image Communication*, v. 56, pp.: 1-11.

DOI: https://doi.org/10.1016/j.image.2017.04.004

Bayesian Adaptive Algorithm for Fast Coding Unit Decision in the High Efficiency Video Coding (HEVC) Standard

Amaya Jiménez-Moreno[a,*], Eduardo Martínez-Enríquez[b], Fernando Díaz-de-María[a]

[a] *Department of Signal Theory and Communications, Carlos III University, Leganés (Madrid), Spain*
[b] *Instituto de Óptica, Consejo Superior de Investigaciones Científicas, Madrid, Spain*

Abstract

The latest High Efficiency Video Coding standard (HEVC) provides a set of new coding tools to achieve a significantly higher coding efficiency than previous standards. In this standard, the pixels are first grouped into Coding Units (CU), then Prediction Units (PU), and finally Transform Units (TU). All these coding levels are organized into a tree-shaped arrangement that allows highly flexible data representation; however, they involve a very high computational complexity.

In this paper, we propose an effective early CU depth decision algorithm to reduce the encoder complexity. Our proposal is based on a hierarchical approach, in which a hypothesis test is designed to make a decision at every CU depth, where the algorithm either produces an early termination or decides to evaluate the subsequent depth level. Moreover, the proposed method is able to adaptively estimate the parameters that define each hypothesis test, so that it adapts its behavior to the variable contents of the video sequences.

The proposed method has been extensively tested, and the experimental results show that our proposal outperforms several methods of the state-of-the-art, achieving a significant reduction of the computational complexity (36.5% and 38.2% average reductions in coding time for two different encoder configurations) in exchange for very slight losses in coding performance (1.7% and 0.8% average bit rate increments).

Keywords: Bayesian statistics; complexity reduction; fast coding unit decision; high efficiency video coding; hypothesis test; *on the fly* estimation.

1. Introduction

1.1. The High Efficiency Video Coding standard

The High Efficiency Video Coding (HEVC) standard is the latest video coding project developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), working together in the Joint Collaborative Team on Video Coding (JCT-VC). An increasing interest in new video applications, higher video resolutions (Full-HD, 4k×2k, or 8k×4k), and the great amount of traffic caused by video signals over the mobile or fixed communication networks were some of the motivations for the design of HEVC, which was developed to serve these new demands. As its predecessors (H.264/AVC or MPEG-2), HEVC is a block-based hybrid video encoder that combines motion-compensated prediction with spatial transform coding, but, by virtue of a set of new coding techniques [1], HEVC is able to double the compression rates for the same objective image quality compared to H.264/AVC [2].

There are many new features included in HEVC. Below, we briefly present those that become more relevant to describe our proposal; specifically:

- The HEVC standard uses a quadtree structure to divide the frame into blocks called Coding Tree blocks (CTBs). The size L×L of a luminance CTB can be chosen as L = 16, 32, or 64 pixels, where larger L values usually results in better compression [1]. This is one of the main differences with respect to previous standards, where typically a size of 16×16 pixels was used.

*Corresponding author
Email addresses: ajimenez@tsc.uc3m.es (Amaya Jiménez-Moreno), eduardo.martinez@io.cfmac.csic.es (Eduardo Martínez-Enríquez), fdiaz@tsc.uc3m.es (Fernando Díaz-de-María)

- HEVC enables to divide the CTBs into smaller coding units (CUs) using a quadtree structure. The root of this tree is associated with the CTB, allowing CU sizes up to the CTB size and down to 8×8 pixels. The higher depth in the quadtree, the lower CU dimensions. Thus, a CTB may consist of only one CU or multiple CUs. Fig. 1 shows an example of a CTB of 64×64 pixels, divided into 13 CUs with dimensions ranging from 32×32 to 8×8 pixels.

- For each CU the encoder must decide whether to use intra- or inter-coding to best represent it. In the prediction stage, each CU can be divided into prediction units (PUs) of several sizes for which either the intra- or the inter-prediction is estimated (depending of the prediction type, several partitioning modes are allowed). Fig. 2 illustrates all the PU partitioning possibilities for a 2N×2N CU. For inter-coding, the PU could have either the same dimensions as the CU, or could be divided into two or four blocks (which could even be of asymmetric sizes). For intra-coding only 2N×2N and N×N sizes are allowed.

- Concerning the transformation of the prediction residual, the CU can be considered as the root of another quadtree of transformation units (TUs). The sizes of the TUs can vary from 4×4 up to the CU size, depending on the depth at which the TU is located. HEVC uses an integer transform similar to the discrete cosine transform (DCT) defined for several sizes from 4×4 up to 32×32. On the left side of the Fig. 3 we show an example where a 64×64 CTB is divided into the corresponding CUs (in solid line) and TUs (dashed line). On the right part of Fig. 3, the associated quadtree is shown.

From all these coding options, the encoder needs to select the best combination to represent each CTB. This process is explained in detail in the next subsection.

1.2. Rate-Distortion Optimization in HEVC

An HEVC encoder selects the best coding tree structure (including the CUs, PUs, and TUs sizes) through a Rate-Distortion Optimization (RDO) process, which must evaluate every tree configuration and compare all of them in terms of rate and distortion. This process contributes to achieve a very high coding efficiency but at the expense of a large increase in the encoder complexity.

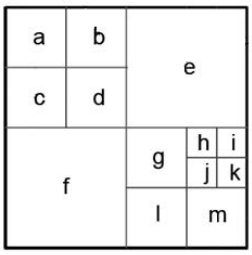


Figure 1: An example of a 64×64 CTB divided into 13 CUs. Two 32×32 CUs (e, and f), seven 16×16 CUs (a, b, c, g, l, and m) and four 8×8 CUs (h, i, j, and k).

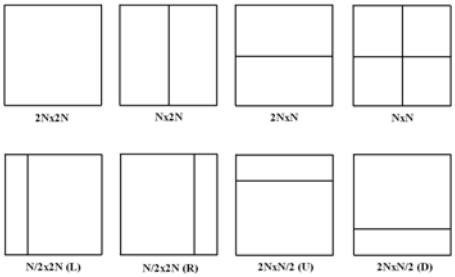


Figure 2: PU sizes allowed for a 2N×2N CU. In the upper part the symmetrical partitions are shown. In the lower part the asymmetrical ones, where L, R, U, and D stand for the position of the smallest partition (left, right, up and down, respectively).

Specifically, the RDO process aims to find the coding option that minimizes a distortion measure subject to a given rate constraint:

$$\min_{\boldsymbol{\theta}} \{D(\boldsymbol{\theta})\} \text{ subject to } R(\boldsymbol{\theta}) \leq R_c, \qquad (1)$$

where $\boldsymbol{\theta}$ is a combination of different coding options (CU, PU, and TU sizes, motion vectors (MVs), reference frames, etc.). $D(\boldsymbol{\theta})$ represents the distortion between the original and the reconstructed block of pixels, $R(\boldsymbol{\theta})$ is the rate needed to encode it, and, finally, R_c represents the rate constraint.

Using Lagrange formulation, this constrained problem can be formulated as an unconstrained one [3]:

$$\min_{\boldsymbol{\theta}} \{J\} \text{ with } J(\boldsymbol{\theta}) = D(\boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta}), \qquad (2)$$

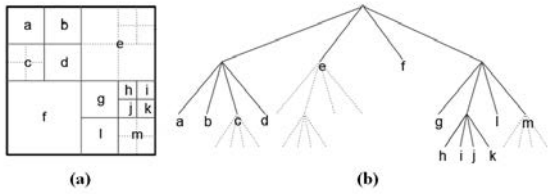


Figure 3: (a) An example of a 64×64 CTB divided into 13 CUs of several sizes in solid line. An example of a TU structure in dashed line where the TU can be either the same as the CU size (a, b, d, f, g, l, h, i, j, and k) or not (c, e, and m). (b) The associated quadtree, in solid line for CUs and dashed line for TUs.

where J represents the rate-distortion (R-D) cost associated with the set of coding options $\boldsymbol{\theta}$, which is calculated as a weighted sum of the Distortion and Rate terms $D(\boldsymbol{\theta})$ and $R(\boldsymbol{\theta})$, where the Lagrange multiplier λ balances the relative importance of both terms. For a given value of λ, the solution $\boldsymbol{\theta}^*(\lambda)$ turns out to be an optimal solution of the original RDO problem in (1) for a particular value of $R_c = R(\boldsymbol{\theta}^*)$.

This solution involves to calculate the total bit rate required to encode each CTB and distortion between original and reconstructed CTBs. Hence, the encoder needs to carry out the prediction, residue calculation, transformation, quantization, entropy coding, and the inverse processes considering each combination of coding options $\boldsymbol{\theta}$. This process results in extremely high computational complexity, being the bottleneck of the HEVC standard.

To alleviate this highly time-consuming process, in this paper we propose a fast method for the selection of the optimal CU depth. The aim of this work is to design an algorithm able to reduce the computational complexity, while maintaining the quality as close as possible to that of the baseline encoder process. The complexity reduction is obtained by limiting the depth of the quadtree at CU level by means of an adaptive Bayesian method, which determines the CTBs where the lower CU depths are likely to be optimal.

The remainder of this paper is organized as follows. In Section 2, an overview of the state-of-the-art methods that address the complexity reduction problem in HEVC is presented along with the main contributions in this paper. Section 3 provides a detailed explanation of our proposal. In Section 4, the experimental evaluation is described and discussed. Finally, Section 5 summarizes our conclusions and outline future lines of research.

2. Review of the state of the art

Several works have been developed to address the problem of the high computational complexity of the HEVC standard. As explained previously, the optimal representation of a CTB is a combination of decisions related to CU depths, PU modes, and TU quadtree sizes; thus, the complexity reduction problem has been approached as an early determination of the CU, PU, or TU.

In this section, a description of some relevant state-of-the-art proposals is provided. Specifically, we focus on the proposals related to the early CU depth determination, which is the objective of this work. Nevertheless, an outline of other methods related to fast PU and TU size decision is also presented.

In [4], a fast CU depth decision was described. By analyzing the CU depths selected in the last frame, the least used CU depths are disabled for the current frame. This decision depends on some thresholds experimentally determined. Then, for every CU, the depths of the neighbor and co-located CUs are analyzed to avoid checking unnecessary CU depths, depending on the number of neighbor CUs that satisfy some requirements.

[5] explained a fast CU depth selection using Bayesian statistics. This algorithm is based on class-conditional probability density functions (*pdfs*) estimated offline and stored in a look up table. Then, using a Bayesian decision rule, the thresholds (to decide whether to check the next CU depth) are calculated offline for different coding settings and different sequence resolutions.

[6] presented a fast CU encoding scheme based on the spatio-temporal encoding parameters of HEVC. This method utilizes spatial encoding parameters such as sample adaptive offset filter data to estimate the texture complexity in a CU partition. Moreover, the temporal complexity is estimated by means of temporal encoding parameters such as MVs or TU sizes. All these parameters were used to design an early CU Skip mode detection and a fast CU split decision methods.

[7] proposed three methods to save complexity: a Skip mode detection, an early CU termination, and a CU skip estimation. The first one determines for each CU whether the Skip mode should be the only

mode tested. The second and third methods aim to decide if either a larger or a smaller CU size should be evaluated, respectively. The three methods are based on *pdfs* of the RD costs and use the Bayes' rule to make the corresponding decisions.

In [8], a pyramid motion divergence method was proposed to early skip some CU depths. First, before coding a frame, the optical flow is estimated for a downsampled version of that frame. Then, for each CU, the pyramid motion divergence is calculated as the variance of the optical flows of the current CU and the CUs of smaller size. Finally, an algorithm based in euclidean distances selects the optimal CU quadtree structure, as CUs with similar pyramid motion divergence tend to use similar partitions.

[9] described an algorithm to decide the optimal CU partitioning in every CTB based on local statistics of the RD costs and motion activity. This method sets several early termination conditions to decide if the splitting process is terminated. Specifically, if the Skip mode is selected as optimal, or if the RD cost is lower than its average value and the MVs are zero, or if the ratio of the RD costs between the current CU and its parent CUs is lower than a threshold, the following CU depths are not evaluated.

[10] explained a machine learning approach based on Support Vector Machines (SVMs) to determine the optimal CU size. Several features are obtained from the video content and the previously coded data to generate the optimal feature subset by a wrapper feature selection algorithm. To derive an accurate model to predict the CU size, differences of RD costs are used as weights in the SVM training procedure, carried out offline, to alleviate the losses in coding performance due to misclassification.

[11] presented a fast CU size decision for 3-D HEVC video encoders. The authors explain that, when two CUs have similar depth values, it is because there are high correlations between their coding information. In this way, they proposed two efficient CU decision methods: first, an early coding level determination based on the spatially neighboring CUs depth values; second, an adaptive mode size decision based on the correlations between the motion characteristics and the depth values.

In [12] a mechanism to adaptively select the CU depth range was proposed. This method determines the optimal CU range using the distribution of CU depths in the same sequence, avoiding to carry out the encoding process with those depths outside the calculated range. Moreover, the depths of neighbor CUs are used to check if the R-D cost calculations at the current CU depth may be further skipped.

In [13] a fast intra mode decision was described. This work presents a novel texture orientation detection algorithm based on the local directional variance computed along a set of co-lines. This variance is measured using a novel Mean Directional Variance (MDV) metric which calculates the cumulative variance along digital lines. According to the dominant gradient detected, a reduced set of directional candidate modes for each PU are selected to be further tested in the RDO stage.

Another technique for fast decision in intra prediction was described in [14]. This method proposes a CU depth prediction and an early CU splitting termination. The CU depth is limited to a narrower predicted range. CU splitting stops when the R-D cost of current CU is below an estimated R-D cost. Both decisions are based on values previously found in the collocated CTBs of the previous frame.

In [15], a method based on histograms of oriented gradients is described. A codebook is built offline by clustering the histograms obtained from training sequences for each CU depth. Then, the optimal depth is selected by comparing the histogram of the current CU with those of the learned codebook.

[16] proposed to extract visual features in a CTB to simplify the intra coding procedure by reducing the quadtree partition depth for each CTB. A measure of the edge strength in a CTB based on the Sobel operator is used to constrain the possible maximum quadtree partition depth of the CTB.

In [17], a complexity control method for HEVC was proposed. Based on the observation that in the same region of consecutive frames some features (motion, texture, etc.) tend to remain unchanged, the maximum CU depths in previous frames are stored to avoid the complete evaluation in the next frames, which are called constrained frames. The complexity control is achieved by calculating the number of constrained frames between regularly coded frames. This number depends on the complexity spent in the encoding process and a prediction of the encoding complexity for the remaining frames. However, this method does not work properly when encoding fast motion video sequences with small target complexities. To solve this drawback, [18] presented an extension of their previous work. This new version of their algorithm estimates the maximum CU depth based on both spatial and temporal correlations observed among CU depths.

Thus, for every CU, the neighbor CUs are also used along with the CUs of previous frames.

Some examples of papers related to the fast selection of PU modes can be found in [19, 20, 21, 22]. Furthermore, methods that address the early determination of the TU quadtree structure are described in [23, 24, 25].

Some of the previous works are not able to adapt their algorithms over time during the coding process (there are methods that depends on fixed thresholds or statistics calculated offline, e.g., [4], [5], [15], and [10]). Some methods are designed taking into account the spatial and temporal correlations between blocks, e.g., [14], and [17], but the necessary statistical analysis appears to be lacking. Moreover, other approaches have difficulty reaching different operating points to obtain several time savings, as [4] and [9].

The contribution of this work is to design a complexity reduction method based on Bayesian statistics that is able to adapt the parameters of the algorithm *on the fly* to fit different types of content. Moreover, our method is designed to have negligible complexity since it only requires to perform simple mathematical calculations. Our proposal is focused on the fast CU depth selection problem, but it could be extended to manage PU or TU early termination problems. Specifically, for each CU depth, we propose to estimate the *pdfs* that model the two possible hypotheses: (i) the considered CU depth is optimal or (ii) the CU depth is not optimal. Based on these *pdfs* and using Bayes' formulation, we obtain a threshold that allows us to decide if the early termination is in order. The *pdfs* and the thresholds are adapted *on the fly* to the changing content of the sequence that is being encoded. Previous work by the authors presented a complexity control method in which early terminations are based on an R-D cost analysis [26]. In this work, we improve the performance of the method proposed in [26] by means of a Bayesian approach, which formalizes the decision problem and is able to manage the cost of wrong decisions in terms of bit rate increments.

3. Proposed method

3.1. An overview

As shown in [26], the encoder selects lower depths with high probability for all QP values for sequences with smooth movement and static regions. In contrast, the probability of selecting lower depths decreases notably when the sequence is more complex. Moreover, the *a priori* probability of lower depths increases with an increasing QP since these lower depths are more adequate for a coarse coding process.

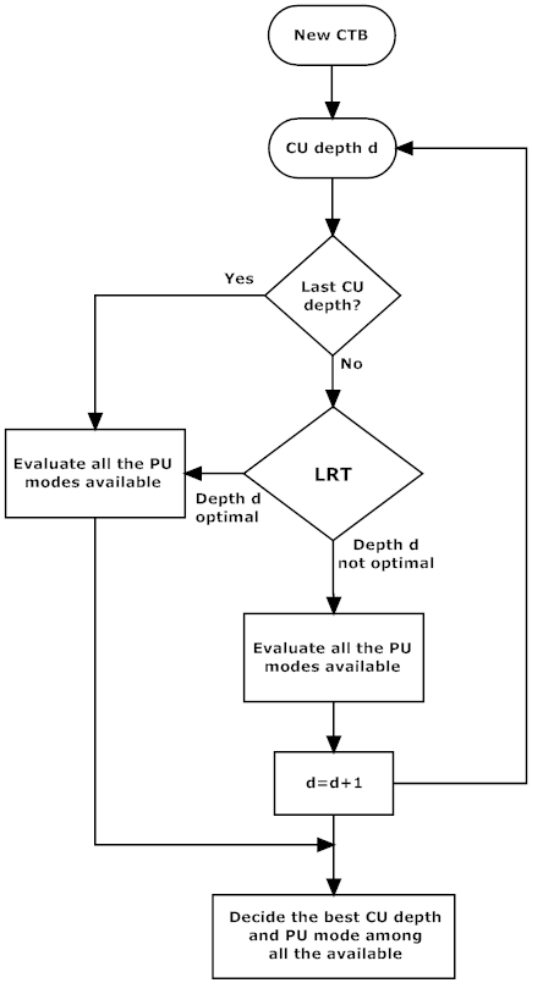


Figure 4: Flowchart of the proposed algorithm.

Given that low CU depths are optimal for several types of sequences and QP values, an appropriate early selection of the CU depth will be effective to reduce the encoder complexity. Furthermore, given that the optimal depth strongly depends on both the particular video content and QP value, an *on the fly* adaptation of the algorithm is needed.

In order to reduce the computational burden of the HEVC standard, the proposed method sets an early termination at each CU depth. Thus, if the

early termination condition is satisfied, the higher CU depths will not be tested, saving the corresponding processing time. On the other hand, if the early termination condition is not met, the next CU depth is evaluated and the following termination condition is checked. Specifically, our algorithm tries to distinguish between two hypotheses: the current CU depth d is the best ($H_0 : depth^* = d$) or the current CU must be divided and higher CU depths must be evaluated ($H_1 : depth^* > d$). This problem can be formulated as a binary hypothesis testing.

Mathematically, using Bayes' formulation, given two possible hypotheses H_0 and H_1 and the two corresponding decisions D_0 and D_1, the likelihood ratio test (LRT) is defined as follows:

$$\frac{\Pr(x|H_1)}{\Pr(x|H_0)} \underset{D_0}{\overset{D_1}{\gtrless}} \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})} \frac{\Pr(H_0)}{\Pr(H_1)}, \qquad (3)$$

where $\Pr(x|H_i)$ is the likelihood of the input feature x given the hypothesis i, $\Pr(H_i)$ is the *a priori* probability of the hypothesis i, and C_{ji} is the cost of deciding j when the correct hypothesis is i.

In our proposal, we build a LRT at every CU depth, where the two possible decisions are: current depth is optimal and the next depths are avoided (D_0); or current depth is not optimal and the following depths need to be evaluated (D_1). The complete process is summarized in Fig. 4.

To correctly develop the LRT, we need to select a suitable input feature x. The feature selection process (that will be explained in detail in the next subsection) will provide us with a proper feature x that, as we will demonstrate, is able to produce two separable *pdfs* for each hypothesis, $\Pr(x|depth^* = d)$ and $\Pr(x|depth^* > d)$. In fact, the more separable *pdfs*, the higher accuracy of the decision. Once these *pdfs* are estimated, we can decide for each new sample, using (3), the most probable hypothesis.

3.2. R-D costs as input features

The encoder decides on the optimal CU depth based on R-D costs in (2). In this section we show that R-D costs can be employed as input feature x in (3) to make accurate early decisions on the CU depth, as proved in [27] and [28] for a similar mode decision problem in H.264/AVC. Let us define the following *pdfs*:

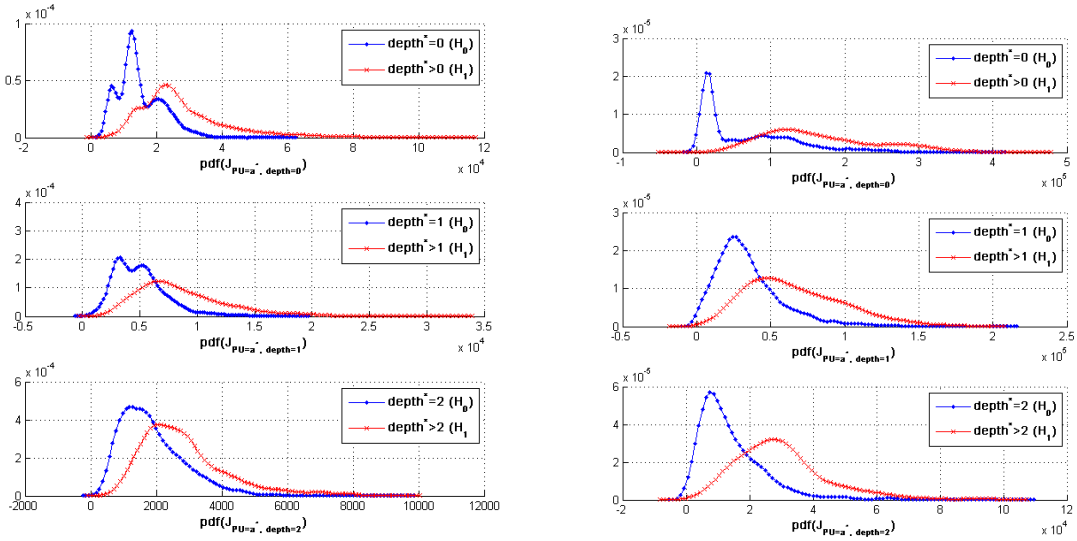
$$\Pr(J_{PU=a,depth=d}|depth^* = d) \text{ and } \Pr(J_{PU=a,depth=d}|depth^* > d), \qquad (4)$$

Table 1: Means and standard deviations of the R-D costs when $depth^* = d$ (H_0) and when $depth^* > d$ (H_1), i.e., $J_{PU=a^*,depth=d}|depth^* = d$ and $J_{PU=a^*,depth=d}|depth^* > d$, respectively, for the sequences "FourPeople" and "Johnny".

			depth=0	depth=1	depth=2
FourPeople	QP 22	$\mu\|H_0$	14918	5191	1890
		$\sigma\|H_0$	6647	2337	977
		$\mu\|H_1$	28508	8923	2904
		$\sigma\|H_1$	14433	4307	1417
	QP 27	$\mu\|H_0$	25360	10065	3983
		$\sigma\|H_0$	14933	5521	2518
		$\mu\|H_1$	58462	19845	7138
		$\sigma\|H_1$	34746	10978	4018
	QP 32	$\mu\|H_0$	47594	21265	8730
		$\sigma\|H_0$	33039	13432	6316
		$\mu\|H_1$	122380	43734	16756
		$\sigma\|H_1$	76016	25384	10391
	QP 37	$\mu\|H_0$	100100	45879	19782
		$\sigma\|H_0$	75318	29843	14695
		$\mu\|H_1$	257150	94998	36921
		$\sigma\|H_1$	147040	50429	20395
Johnny	QP 22	$\mu\|H_0$	11208	5650	1788
		$\sigma\|H_0$	5410	2848	795
		$\mu\|H_1$	27151	7859	2405
		$\sigma\|H_1$	10365	2763	965
	QP 27	$\mu\|H_0$	19939	9725	3209
		$\sigma\|H_0$	14921	4451	1863
		$\mu\|H_1$	46256	14777	5675
		$\sigma\|H_1$	18118	6867	3085
	QP 32	$\mu\|H_0$	35097	17611	6828
		$\sigma\|H_0$	30322	9367	4918
		$\mu\|H_1$	85499	31890	13530
		$\sigma\|H_1$	37488	16101	6316
	QP 37	$\mu\|H_0$	67871	34976	14485
		$\sigma\|H_0$	64297	23071	11140
		$\mu\|H_1$	165870	66632	30196
		$\sigma\|H_1$	76736	32511	14740

where $J_{PU=a,depth=d}$ is the R-D cost associated with the PU partition size a at depth d. We check whether the R-D costs leads to *pdfs* in (4) that are separable enough to apply the LRT formulation and determine the optimal CU depth. Specifically, we analyze the statistics (means and standard deviations) of the J cost associated with the best PU mode a^* at a CU depth d when that depth is optimal and when it is not ($J_{PU=a^*,depth=d}|depth^* = d$ and $J_{PU=a^*,depth=d}|depth^* > d$, respectively).

To gather the data, we used the HM13.0 software [29] with the configuration file "encoder_lowdelay_P_main" (for which the CU size can vary from 64×64 to 8×8 pixels, and all the regular PU modes and TU sizes are available). A subset of the test sequences recommended in [30] were encoded with QP values of 22, 27, 32, and 37. In



(a) "FourPeople" at QP 22

(b) "Johnny" at QP 37

Figure 5: An illustration of the statistical behavior of the *pdfs* at several CU depths for two sequences.

Table 1 we show the results for two of these sequences ("FourPeople" and "Johnny") for the considered QP values and the CU depths 0, 1, and 2 (depth 3 is the highest one and does not require to set up an early termination). $\mu|H_i$ denotes the mean value when the hypothesis i is fulfilled (H_0 means the current depth is optimal, while H_1 means the optimal depth is higher), and $\sigma|H_i$ denotes the standard deviation. To gain insight into the obtained *pdfs*, Fig. 5 represents some examples for depths 0, 1, and 2 for the sequences "FourPeople" at QP 22 and "Johnny" at QP 37, using the same coding configuration mentioned before.

Considering the results presented in Table 1 and Fig. 5, we can conclude that the statistics of each hypothesis exhibit a very different behavior at every CU depth since both mean and standard deviation clearly grow when H_1 is fulfilled. Similar results were obtained for all the analyzed sequences.

In summary, the *pdfs* defined in (4), based on the $J_{PU=a^*,depth=d}$ cost, are suitable to address the CU depth decision problem. Moreover, it should be noted that the statistics highly depend on the QP value and the sequence. Thus, it seems reasonable to use an adaptive algorithm able to manage these changes in the video content and the coding process.

3.3. Studying the Gaussianity of the data

To apply the hypothesis test it is necessary to know the *pdfs* of the data. Our first intuition, from the examples in Fig. 5 and the similar results obtained for all the analyzed sequences, is that the majority of these *pdfs* could be assumed as Gaussian. To check the goodness of this assumption, we have carried out a Kolmogorov-Smirnov test [31], which is commonly used to judge whether two distributions are different at certain significance level ($p-value$). This test builds the cumulative distribution functions (*cdfs*) from the data ($F(x)$) and the standard Gaussian distribution ($G(x)$). Then, the test uses the maximum difference between both *cdfs* to check if the data could be Gaussian. Mathematically, this can be written as:

$$D_{max} = \max(|F(x) - G(x)|). \qquad (5)$$

The low $p-value$ is induced by the large value of D_{max} and indicates that difference between both distributions is significant. The Kolmogorov-Smirnov test takes the decision between Gaussian data ($H_{0,KS}$=1, accepting the null hypothesis) or not ($H_{0,KS}$=0, rejecting the null hypothesis) based on the $p-value$. The higher $p-value$, the more likely is the null hypothesis. In our case, we have applied this test with a $p-value$ of 5%.

Table 2: The Kolmogorov-Smirnov test results for the sequence "FourPeople" at QP 22.

			depth=0	depth=1	depth=2
QP 22	$d^* = d$	$H_{0,KS}$	1	0	0
		D_{max}	0.13	0.07	0.07
		$p-value$	0.02	0.08	0.10
	$d^* > d$	$H_{0,KS}$	1	0	0
		D_{max}	0.14	0.09	0.09
		$p-value$	0.03	0.20	0.07

Table 3: The Kolmogorov-Smirnov test results for the sequence "Johnny" at QP 37.

			depth=0	depth=1	depth=2
QP 37	$d^* = d$	$H_{0,KS}$	1	0	0
		D_{max}	0.25	0.10	0.11
		$p-value$	$8.91e-8$	0.11	0.10
	$d^* > d$	$H_{0,KS}$	0	0	0
		D_{max}	0.09	0.08	0.12
		$p-value$	0.14	0.36	0.09

To carry out the test, we used the same procedure and subset of sequences as in the previous section and, for illustrative purposes, we show the obtained results for the same sequences as in Fig. 5. In particular, Tables 2 and 3 show several results for the sequences "FourPeole" and "Johnny", respectively. For each sequence we measure the gaussianity of the *pdfs* $J_{PU=a^*,depth=d}|depth^* = d$ (labeled as $d^* = d$ in the tables) and $J_{PU=a^*,depth=d}|depth^* > d$ (labeled as $d^* > d$) in every depth. For both *pdfs* we show $H_{0,KS}$, D_{max}, and $p-value$.

As it can be seen from the results, the majority of the RD cost data can be derived from Gaussian distributions and only some specific examples reject this hypothesis. To give a graphical example, we represent several of the obtained *cdfs* in Fig. 6. Specifically, in each figure the Gaussian *cdf* is plotted along with the *cdf* associated to the data. The figure on the left corresponds to "FourPeople" at QP 22 with $depth = 0|depth^* = 0$; as it can be seen, there are no high similarity between both *cdfs*, so the Gaussiniaty hypothesis is rejected. In the center and the right part of the figure (with data coming from "FourPeople" at QP 22 with $depth = 3|depth^* = 3$ and "Johnny" at QP 37 with $depth = 2|depth^* > 2$, respectively), the Gaussianiaty hypothesis is accepted as the data *cdfs* are very similar to the Gaussian distribution.

It must be noted that similar results, endorsing the Gaussianity assumption, are obtained with most of the sequences evaluated.

3.4. On the fly estimation of the pdfs

Once we have defined the hypothesis H_0 and H_1, selected the input feature $x = J_{PU=a^*,depth=d}$, and proved the Gaussianity of the data, this section is devoted to explain the estimation of the statistical parameters that define the *pdfs* in the LRT formulation in (3).

By virtue of the Gaussian assumption proved in the previous section, we only need to store the means and the standard deviations to fully describe the *pdfs*. In this way, the hypothesis test in (3) can be written as:

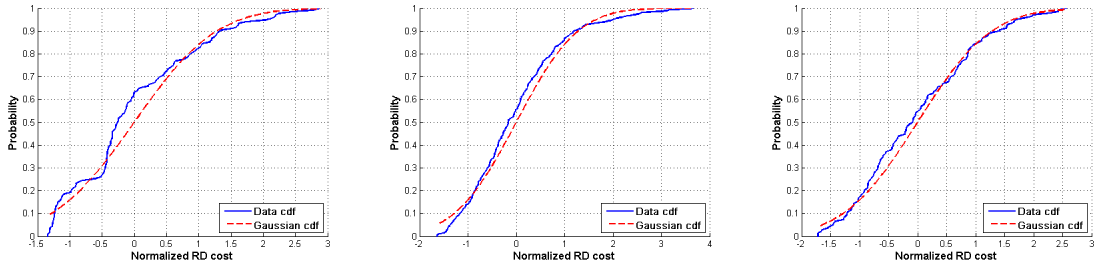
$$-\frac{(J_{2N\times 2N,0}-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2}+\frac{(J_{2N\times 2N,0}-\hat{\mu}_0)^2}{2\hat{\sigma}_0^2}+ \\ +\ln\frac{\hat{\sigma}_0^2}{\hat{\sigma}_1^2}\gtrless_{D_0}^{D_1}\ln(\frac{\hat{P}(H_0)}{\hat{P}(H_1)})+\ln(\frac{C_{10}}{C_{01}}), \quad (6)$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ are the estimated mean and standard deviation of $P_r\left(J_{PU=a^*,depth=d}|depth^* = d\right)$; $\hat{P}(H_i)$ is the estimated *a priori* probability of hypothesis i; and the cost associated with correct decisions (C_{00} and C_{11}) have been considered to be zero.

We propose to estimate *on the fly* the parameters $\hat{\mu}_0$, $\hat{\mu}_1$, $\hat{\sigma}_0$, and $\hat{\sigma}_1$, and the *a priori* probabilities, $\hat{P}(H_0)$ and $\hat{P}(H_1)$ to adapt them to the content along the coding process. Two different procedures to carry out the estimation were studied, specifically, an arithmetic and an exponential average.

In Fig. 7 an illustrative example of the results achieved by the arithmetic and the exponential averages is shown. In particular, we show the $\hat{\mu}_0$ and $\hat{\mu}_1$ mean values calculated over 350 samples with both kinds of averages. Furthermore, the expected mean values per 50-sample block are also shown as a reference to evaluate the tracking ability of the compared estimation methods. As observed in Fig. 7, the exponential average produces better tracking than the arithmetic, achieving a mean value closer to the expected value, while the arithmetic average gets stuck as the number of samples grows. Another interesting aspect that can be analyzed in Fig. 7 is the difference between the mean values of both *pdfs*. As can be observed, $\hat{\mu}_0$ and $\hat{\mu}_1$ are clearly different, which helps the LRT to take more accurate decisions.

According to this analysis, we use an exponential average for the estimation of the *pdfs* parameters



(a) (b) (c)

Figure 6: Three examples of the resulting *cdfs* to carry out the Kolmogorov-Smirnov test (a) "FourPeople"at QP 22 with $depth = 0|depth^* = 0$; (b) "FourPeople" at QP 22 with $depth = 3|depth^* = 3$; and (c) "Johnny" at QP 37 with $depth = 2|depth^* > 2$.

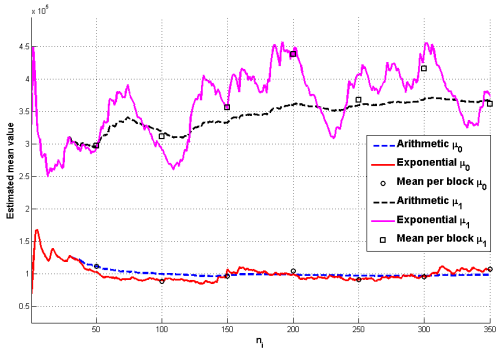


Figure 7: Arithmetic and exponential estimations of $\hat{\mu}_0$ and $\hat{\mu}_1$ in BasketballDrill at QP 32 for $depth = 0$.

that relies on the following updating equations:

$$\hat{\mu}_i(n_i) = \alpha\hat{\mu}_i(n_i - 1) + \\ +(1-\alpha)\cdot J_{PU=a^*,depth=d}(n_i), \\ \text{with } i = \{0,1\} \quad (7)$$

$$\hat{\sigma}_i^2(n_i) = \alpha\hat{\sigma}_i^2(n_i - 1) + \\ +(1-\alpha)\cdot (J_{PU=a^*,depth=d}(n_i) - \hat{\mu}_i(n_i))^2, \\ \text{with } i = \{0,1\}, \quad (8)$$

where n_i is an index associated with the times that the hypothesis H_i is selected; $\hat{\mu}_i(n_i - 1)$ and $\hat{\mu}_i(n_i)$ are the estimated means at the instants $(n_i - 1)$ and n_i, respectively (the notation is the same for the variances $\hat{\sigma}_i^2(n_i - 1)$ and $\hat{\sigma}_i^2(n_i)$); $J_{PU=a^*,depth=d}(n_i)$ is the cost at the instant n_i; and α is the parameter defining the forgetting factor of the exponentially averaged estimation process (α has been set to 0.95 in our experiments).

The *a priori* probabilities, $\hat{P}(H_0)$ and $\hat{P}(H_1)$, are calculated by counting the number of occurrences of every hypothesis. Additionally, to avoid the *winner-takes-all* problem, we limit the maximum possible value of $\hat{P}(H_0)$ and $\hat{P}(H_1)$.

It should be noted that the parameter $\ln(C_{10}/C_{01})$ in (6) allows us to manage the trade-off between time saving and bit rate increment of our method. If $\ln(C_{10}/C_{01})$ takes a high value, the early termination (D_0) is selected more often; therefore, the time savings are higher at the expense of higher increments in the final bit rate, and vice-versa. To reach a suitable operating point, we carried out an experimental validation of this parameter, following the same procedure as in Section 4, with a subset of the sequences presented in Table 4; specifically, "BasketballDrill", "BasketballPass", "BlowingBubbles", "BQTerrace", "FourPeople", and "Johnny". We obtain the following values for the three decision levels: depth 0: $\ln(C_{10}/C_{01}) = -2$; depth 1: $\ln(C_{10}/C_{01}) = -2$; and depth 2: $\ln(C_{10}/C_{01}) = -1$.

Regarding the computational complexity, we have carefully studied the floating-point operations involved and proposed an alternative integer-based implementation. Specifically, our proposal is comprised of two changes: 1) since both the statistical parameters and the threshold (6) come from RD costs that tend to take high values, the decimals turn out to be irrelevant; thus, the calculation of the exponential averages of the means and standard deviations ((7) and (8)) can be carried out by scaling and rounding the values to avoid the decimals; 2) in (6), the logarithm of a ratio can be expressed as the difference of the logarithms; thus, we avoid computing the divisions while the logarithms can be

implemented by means of a look up table. Finally, the two divisions of the left hand side of (6) are implemented as floating-point operations; next, the results can be scaled and rounded to complete the calculation by means of integer-based operations.

3.5. Summary of the algorithm

The complete method is summarized in Algorithm 1.

Algorithm 1 Proposed coding process.

Require: M: number of CTBs in a frame
Require: D: number of available CU depths in a CTB

```
for ∀m ∈ M do
  for ∀d ∈ D do
    Evaluate all PU partition modes in depth d
    Calculate the input feature to the hypothesis testing J_{PU=a*,depth=d}
    Apply the hypothesis testing in (6)
    if H_0 is selected then
      Go to 10
    end if
  end for
  Decide the best coding options among all CU depths and PU partitions evaluated
end for
```

4. Experiments and results

4.1. Experimental setup

To properly test the performance of our proposal, it was integrated into the HEVC reference software HM13.0 [29]. The experimental setup followed the recommendations given in [30]. Specifically, we used the suggested QP values (22, 27, 32, and 37) and the set of sequences listed in Table 4. Moreover, the configuration files chosen for this evaluation were "encoder_lowdelay_P_main" and "encoder_randomaccess_main". To measure the coding performance, the bit rate increment (ΔBR) was calculated following the guidelines given in [32]. To measure the encoder complexity, the time saving (TS) was calculated as follows:

$$TS = \frac{Time(HM13.0) - Time(Proposed)}{Time(HM13.0)} \times 100. \quad (9)$$

4.2. Results and comparison with other state-of-the-art approaches

Our proposal was compared with several methods of the state-of-the-art: [4], [7], [9], and [26]. The first three are complexity reduction methods addressing the early CU decision problem, as our proposal. As explained before, [4] uses the information related to the optimal CUs in previous frames and neighboring CTBs to reduce the number of depths to be tested. [7] defines a threshold based on statistics of the *pdfs* of the RD costs to design a CU depth early termination process and a Skip mode detection algorithm. To carry out a fair comparison, our implementation of this method has been limited to the CU decision stage. [9] defines several decision stages to decide the optimal CU depth based on local statistics of the RD costs, the optimal PU mode, and the MVs. Last, [26] is a complexity control method based on statistical information that aims to select the CTBs where an early CU decision is beneficial, given a target complexity.

Table 4 shows the results obtained with [4], [7], [9], [26] and the proposed method, for the Low Delay configuration, in terms of ΔBR and TS, averaged across the four considered QP values (22, 27, 32, and 37). The last two rows in the table show the average and the standard deviation over all the sequences. Moreover, to illustrate the behavior of each compared method for some representative sequences, the corresponding R-D curves are plotted in Fig. 8, where the performance of the HEVC reference software HM13.0 is also provided for reference.

As can be seen, our proposal achieves large complexity savings at the expense of limited losses in coding performance. Specifically, we achieve an average TS of 36.5%, reaching a TS near to or higher than 50% in up to six sequences. In coding efficiency terms, our method incurs an average ΔBR of only 1.69%, and only one sequence produces a ΔBR above 3%.

Regarding the compared methods, our proposal clearly outperforms [4], [9], and [26], and achieves similar average results than [7]. In particular, the average ΔBR is notably lower for our method when compared with [4], obtaining at the same time a significant gain in terms of TS. As can be seen, the TS achieved by [4] in many sequences is above 30%; however, this is at the expense of high losses in coding efficiency, especially in sequences containing fast motion or texture. This is due to the fact that

Table 4: Comparative performance evaluation vs. [4], [7], [9], and [26] with Low Delay configuration.

	[4]		[7]		[9]		[26]		Proposed Method	
Sequence	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$
BasketballPass (416×240)	4.76	33.07	1.33	23.92	4.02	30.44	2.64	27.93	2.60	33.57
BlowingBubbles (416×240)	3.08	32.88	1.42	17.83	4.30	24.88	7.33	34.72	1.28	18.85
BQSquare (416×240)	2.52	31.40	1.82	16.14	7.15	29.81	8.06	31.36	1.82	19.14
RaceHorses (416×240)	1.17	34.78	0.56	8.32	3.01	11.97	10.37	30.91	1.81	16.35
Goonies (720×432)	6.70	28.50	1.50	32.74	1.80	38.37	0.39	27.26	0.01	30.77
Spiderman (720×576)	3.18	28.38	0.82	17.90	3.58	24.52	5.69	29.31	0.36	17.54
BasketballDrill (832×480)	6.49	33.45	0.16	10.36	7.85	39.74	3.28	34.06	3.12	38.90
BasketballDrillText (832×480)	6.31	35.17	0.36	10.66	8.44	38.66	3.90	34.97	2.92	37.29
BQMall (832×480)	4.64	35.76	4.56	30.79	9.83	35.92	4.88	36.35	2.20	28.88
PartyScene (832×480)	2.01	35.05	0.57	7.31	7.24	25.18	18.81	45.92	2.22	20.96
ChinaSpeed (1024×768)	3.41	36.53	0.29	15.84	8.11	31.55	10.56	34.68	2.73	30.15
FourPeople (1280×720)	10.16	20.53	1.86	57.54	5.32	65.70	0.46	36.68	1.02	49.89
Johnny (1280×720)	9.59	18.59	1.31	57.73	4.66	65.71	−0.23	32.20	0.51	50.80
KristenAndSara (1280×720)	21.39	25.43	1.06	55.72	3.78	64.30	0.35	33.99	0.49	45.47
SlideEditing (1280×720)	11.63	7.55	0.42	39.58	10.85	77.86	0.80	36.39	2.40	55.81
SlideShow (1280×720)	7.05	21.10	1.46	40.78	14.23	58.27	3.07	28.90	1.91	44.66
Vidyo1 (1280×720)	20.25	28.13	1.36	59.04	3.00	65.56	0.04	32.97	0.89	50.05
Vidyo3 (1280×720)	14.79	30.82	1.70	56.15	3.78	61.11	−0.13	31.34	2.40	58.98
Vidyo4 (1280×720)	6.38	21.72	0.97	45.48	4.26	58.79	0.24	34.85	1.44	47.64
BasketballDrive (1920×1080)	13.81	35.27	0.62	30.47	3.14	41.95	1.13	30.94	1.67	41.89
BQTerrace (1920×1080)	10.09	33.16	2.77	38.81	6.68	49.51	4.01	41.33	1.72	35.22
Cactus (1920×1080)	5.49	30.62	1.93	38.72	5.81	46.98	1.41	30.41	1.87	40.46
Kimono (1920×1080)	3.80	22.09	0.57	27.94	2.25	45.72	0.85	33.51	1.56	29.09
ParkScene (1920×1080)	8.00	35.04	0.31	23.25	5.29	42.63	2.50	35.36	1.75	33.49
Average	**7.77**	**28.95**	**1.23**	**31.79**	**5.76**	**44.79**	**3.76**	**33.59**	**1.69**	**36.49**
Standard deviation	**5.39**	**7.17**	**0.96**	**17.15**	**3.00**	**16.84**	**4.55**	**4.14**	**0.83**	**12.49**

[4] makes decisions based on fixed thresholds, which prevent the algorithm from adapting to a variety of sequences with different properties.

[9] achieved higher TS, but in exchange for incurring relevant losses in coding performance. When we compare some specific sequences where the TS is similar, e.g., "BasketballDrill", "BasketballPass", or "ChinaSpeed", we found that our proposal clearly achieves better coding performance, generating approximately half of ΔBR.

The results of [7] reveal a very similar performance than that of our method. The slight increment in the TS obtained with our proposal also generates a slight increment in ΔBR. It is also worth mentioning that the standard deviation of the TS results is significantly lower for the proposed method (12.49 vs. 17.15); i.e., the results seem to be less dependent of the sequence. In particular, we found that [7] has difficulty to achieve relevant TS in some sequences, e.g., "RaceHorses", "BasketballDrill", or "PartyScene".

To fairly compare our method with [26], we configured [26] to produce a similar average TS than that of our proposal. Under these conditions, the

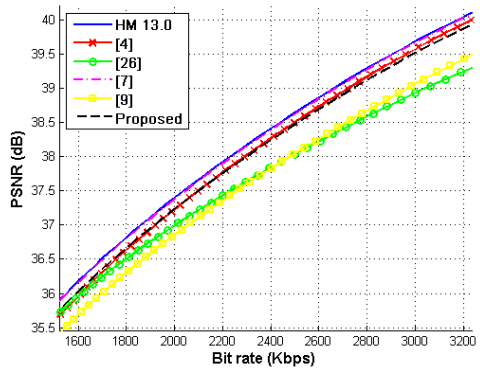
proposed method incurs a ΔBR lower than that of [26] (1.69% vs. 3.76%).

The results for "Goonies" and "Spiderman" deserve a comment. These sequences were included in this test because they are excerpts from real movies that contain several shot changes and, therefore, allow us to test the ability of the proposed method to adapt to new content. Specifically, each one of the selected excerpts show two shot changes. Any non-adaptive method, such as [4] or [9], generates high losses in coding performance. Averaging the results for these two sequences, [4] incurred a 4.9% of ΔBR and [9] a 2.7%, while our proposal only generated a 0.2%, for a similar average TS. When compared to [7], the proposed method also performs significantly better, incurring an average 0.18% ΔBR vs. a 1.16% ΔBR of [7] for similar values of TS. This improvement is due to the fact that our method uses an *on the fly* estimation of the statistical parameters in the hypothesis test, which allows a very fast and accurate adaptation to the new video content.

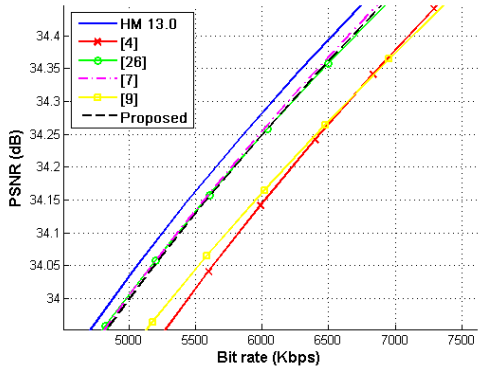
The proposed method was also tested for the Random Access configuration, keeping the rest of the configuration parameters unaltered. To this end, since the statistics of P and B frames are different [27], the statistical parameters of our method were estimated separately for each kind of frame. In this case only [7] was taken as reference for comparisons since the other methods ([4], [9], and [26]) achieved notably lower performances in previous experiments. The results are shown in Table 5. As it can be seen, they follow the same pattern as for the Low Delay configuration, and the compared methods provide very similar performances. In particular, the proposed method obtains a slight increment in the TS, but also in ΔBR, though the losses in coding efficiency are more irrelevant than that with the Low Delay configuration. Furthermore, again, the standard deviation of the results in terms of TS is significantly lower for the proposed method (11.80 vs. 17.10); in particular, [7] fails to reach competitive TSs for several sequences, e.g., "RaceHorses", "PartyScene" or "SlideEditing".

With these results we have demonstrated that our proposal is able to properly work with different kinds of frames and to adapt its behavior to varying content.

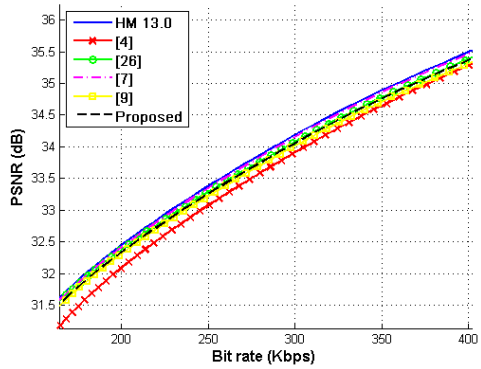
In summary, our method obtains a superior performance compared to [4] and [9] by virtue of its statistical basis and the *on the fly* estimation of the *pdf* parameters, which allows a proper adaptation to the content of the sequences. The proposed method also improves the performance of [26] thanks to the Bayesian approach, which takes into account the distribution of the two possible hypotheses. This is also a conceptual advantage with respect to [7] that



(a)

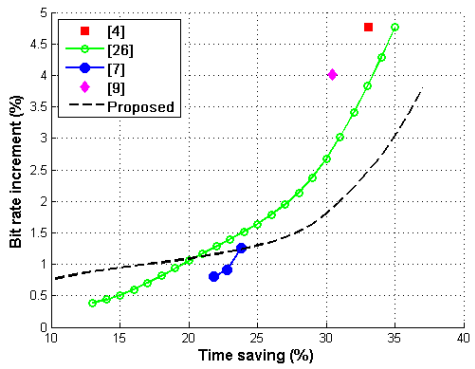


(b)

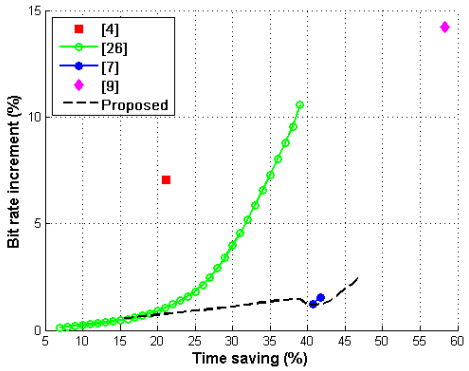


(c)

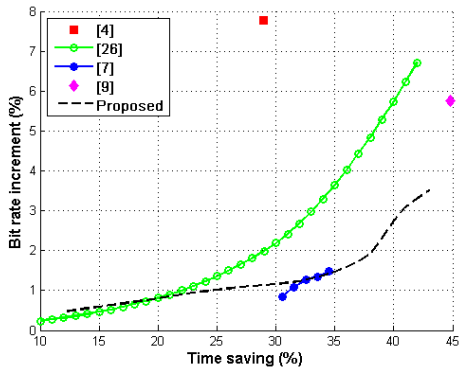
Figure 8: R-D performance for representative sequences: (a) "ChinaSpeed"; (b) "BQTerrace"; (c) "BasketballPass".



(a)



(b)



(c)

Figure 9: Performance evaluation of the proposed method in comparison to [4], [7], [9], and [26]. The graphs show bit rate increment as a function of the computational time saving for (a) "BasketballPass"; (b) "SlideShow"; and (c) over all the considered sequences (average).

could explain the lower standard deviation of the TS results with our method. Moreover, although the results obtained with the proposed method and [7] are very similar, our method overcomes one of the weaknesses of [7], which is not able to reach different operating points, as we show in the next section.

Table 5: Comparative performance evaluation vs. [7] with Random Access configuration.

	[7]		Proposed Method	
Sequence	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$
BasketballPass (416×240)	0.49	33.56	0.46	37.63
BlowingBubbles (416×240)	0.71	19.29	0.75	22.68
BQSquare (416×240)	0.36	28.30	0.34	22.38
RaceHorses (416×240)	0.27	9.74	0.53	15.77
Goonies (720×432)	0.58	33.72	0.12	33.93
Spiderman (720×576)	13.10	8.09	5.06	16.50
BasketballDrill (832×480)	0.79	14.28	0.91	36.77
BasketballDrillText (832×480)	0.67	16.55	1.06	36.85
BQMall (832×480)	1.03	32.99	1.18	36.26
PartyScene (832×480)	0.08	1.60	1.59	28.35
ChinaSpeed (1024×768)	1.16	29.61	2.02	36.56
FourPeople (1280×720)	0.12	60.88	0.09	55.72
Johnny (1280×720)	0.27	35.28	−0.04	37.07
KristenAndSara (1280×720)	0.33	54.50	0.13	45.32
SlideEditing (1280×720)	0.38	9.45	0.75	59.21
SlideShow (1280×720)	1.75	21.08	0.38	50.31
Vidyo1 (1280×720)	0.40	59.29	0.41	51.94
Vidyo3 (1280×720)	0.30	56.11	0.62	56.35
Vidyo4 (1280×720)	0.17	47.26	0.23	43.75
BasketballDrive (1920×1080)	0.44	40.01	0.54	44.28
BQTerrace (1920×1080)	0.68	46.76	0.48	38.01
Cactus (1920×1080)	0.68	40.07	0.82	41.64
Kimono (1920×1080)	0.79	38.02	0.55	29.93
ParkScene (1920×1080)	0.83	41.05	0.80	39.89
Average	**1.09**	**32.39**	**0.82**	**38.21**
Standard deviation	**2.58**	**17.10**	**1.01**	**11.80**

4.3. Results in different operating points

[26] was developed for complexity control purposes, while the complexity reduction proposals

are not designed to meet a given target complexity. Nevertheless, our design can produce different operating points in terms of TS by adjusting the term $\ln(C_{10}/C_{01})$ in (6). In this manner, we can manage the trade-off between TS and ΔBR. For example, if we needed large computational savings, $\ln(C_{10}/C_{01})$ would take a high value; or if we needed to be careful with the increments in bit rate, $\ln(C_{10}/C_{01})$ would be lower.

To illustrate this capability, Fig. 9 shows the performance achieved by our proposal for different values of $\ln(C_{10}/C_{01})$, i.e., for different trade-offs between coding complexity (TS, horizontal axis) and increment in bit rate (ΔBR, vertical axis). Parts (a) and (b) of the figure show the results achieved for two representative sequences, "BasketballPass" and "SlideShow", respectively. Part (c) shows average results over all the considered sequences. Moreover, the performance obtained by [4], [9], [7], and [26] are also shown for comparison purposes. Specifically, several TS values are produced with [26] to compare its performance with that of our proposal in several operating points. However, [4] and [9] are only able to generate one operating point because they are not designed to obtain varying performance.

Regarding [7], there is free parameter, originally denoted as θ, that apparently could provide different trade-offs between TS and bit rate increment. However, we have found that this method is actually not able to reach different operating points by acting upon θ, achieving very similar TSs in all our tests.

As can be seen, our proposal clearly outperforms [4] and [9] for the same TS operating point. Compared with [7], the main difference is that the proposed method is able to reach a wide range of TS-ΔBR trade-offs, while [7] is limited to operate in a very narrow range. Compared with [26], for low complexity reductions the performances of both methods are very similar, although sometimes favorable to [26]. However, when higher complexity reductions are required, the results of the proposed method clearly improve those of [26].

4.4. Contribution of each decision level

Finally, we assess the contribution of each decision level to the total performance of the proposed algorithm. As our proposal sets an early termination at every CU depth, we evaluate the influence of each one as a function of the content of the video sequence. In Table 6 we present the results for a reduced set of sequences, which exhibit different properties, when the algorithm is allowed to perform early terminations at the first CU depth ($d = \{0\}$), at the first two CU depths ($d = \{0, 1\}$), or at three possible decision levels ($d = \{0, 1, 2\}$).

As can be seen, the behavior heavily depend on the sequence. Specifically, in those that either are static or exhibit smooth and constant movement, such as "BasketballDrill", and "FourPeople", all the CU depths contribute to the encoding process and our algorithm achieves significant benefits at each decision level. However, in sequences exhibiting either higher level of detail or fast movement, such as "BlowingBubbles" and "RaceHorses", the first depths are not used often and our method starts making early decisions at the highest depths. Summarizing, our method is able to adaptively distribute the early terminations over the possible depth levels according to the complexity of the sequence.

5. Conclusions and further work

In this work we have proposed a CU depth decision algorithm for the HEVC video coding standard. The proposed method is based on the statistical analysis of the CU depths used by the reference encoder, according to which it performs a hypothesis test to decide, at every CU depth, if the following CU depths must be evaluated. Moreover, we adapt each hypothesis test based on *pdfs* estimated *on the fly*, which allows the algorithm to adapt over the variable content of the sequence.

The proposed method has been extensively tested and the experimental results have shown that it works very efficiently for a great variety of sequences, outperforming the results achieved by several methods of the state-of-the-art and achieving either 36.5% reduction in encoding time with just 1.7% bit rate increment for the Low Delay configuration or 38.2% time saving in exchange for 0.8% bit rate increment for the Random Access configuration, in both cases, with respect to the HEVC reference software HM13.0.

An interesting future line of work would be to extend our design to the fast selection of the best PU mode or the best TU structure, which would provide additional computational savings. Moreover, we could also study the use of more complex classifiers to achieve a higher accuracy in the decision process at each level.

Table 6: Evaluation of the contribution of each decision level.

	$d = \{0\}$		$d = \{0, 1\}$		$d = \{0, 1, 2\}$	
Sequence	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$	$\Delta BR(\%)$	$TS(\%)$
BlowingBubbles (416×240)	−0.03	0.18	0.23	1.44	1.28	18.85
RaceHorses (416×240)	0.00	0.77	−0.05	1.94	1.81	16.35
BasketballDrill (832×480)	1.54	19.44	2.03	29.24	3.12	38.90
BQMall (832×480)	0.37	5.33	0.85	15.80	2.20	28.88
ChinaSpeed (1024×768)	0.32	6.56	1.29	21.58	2.73	30.15
FourPeople (1280×720)	0.57	25.38	0.79	39.00	1.02	49.89
Cactus (1920×1080)	0.66	20.33	1.17	29.77	1.87	40.46

6. Acknowledgments

This work has been partially supported by the National Grant TEC2014-53390-P of the Spanish Ministry of Economy and Competitiveness.

References

[1] G. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) Standard, Circuits and Systems for Video Technology, IEEE Transactions on 22 (12) (2012) 1649–1668. doi:10.1109/TCSVT.2012.2221191.

[2] J. Vanne, M. Viitanen, T. Hamalainen, A. Hallapuro, Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs, Circuits and Systems for Video Technology, IEEE Transactions on 22 (12) (2012) 1885–1898. doi:10.1109/TCSVT.2012.2223013.

[3] H. Everett, Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources, Operations Research 11 (3) (1963) 399–417.

[4] J. Leng, L. Sun, T. Ikenaga, S. Sakaida, Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC, in: Multimedia and Signal Processing (CMSP), 2011 International Conference on, Vol. 1, 2011, pp. 56–59. doi:10.1109/CMSP.2011.167.

[5] X. Shen, L. Yu, J. Chen, Fast coding unit size selection for HEVC based on Bayesian decision rule, in: Picture Coding Symposium (PCS), 2012, 2012, pp. 453–456. doi:10.1109/PCS.2012.6213252.

[6] S. Ahn, B. Lee, M. Kim, A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding, IEEE Transactions on Circuits and Systems for Video Technology 25 (3) (2015) 422–435. doi:10.1109/TCSVT.2014.2360031.

[7] J. Lee, S. Kim, K. Lim, S. Lee, A Fast CU Size Decision Algorithm for HEVC, IEEE Transactions on Circuits and Systems for Video Technology 25 (3) (2015) 411–421. doi:10.1109/TCSVT.2014.2339612.

[8] J. Xiong, H. Li, Q. Wu, F. Meng, A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence, Multimedia, IEEE Transactions on 16 (2) (2014) 559–564. doi:10.1109/TMM.2013.2291958.

[9] K. Goswami, B. Kim, D. Jun, S. Jung, J. Choi, Early Coding Unit-Splitting Termination Algorithm for High Efficiency Video Coding (HEVC), ETRI Journal on Information, Telecommunications and Electronics 36 (3) (2014) 407–417. doi:http://dx.doi.org/10.4218/etrij.14.0113.0458.

[10] X. Shen, L. Yu, CU splitting early termination based on weighted SVM, EURASIP Journal on Image and Video Processing 2013 (1) (2013) 1–11. doi:10.1186/1687-5281-2013-4.

[11] Q. Zhang, X. Wang, X. Huang, R. Su, Y. Gan, Fast mode decision algorithm for 3D-HEVC encoding optimization based on depth information, Digital Signal Processing 44 (2015) 37 – 46. doi:http://dx.doi.org/10.1016/j.dsp.2015.06.005.

[12] Y.-F. Cen, W.-L. Wang, X.-W. Yao, A fast CU depth decision mechanism for HEVC, Information Processing Letters 115 (9) (2015) 719 – 724. doi:http://dx.doi.org/10.1016/j.ipl.2015.04.001.

[13] D. Ruiz, G. Fernandez-Escribano, J. L. Martinez, P. Cuenca, Fast intra mode decision algorithm based on texture orientation detection in HEVC, Signal Processing: Image Communication 44 (2016) 12 – 28. doi:http://dx.doi.org/10.1016/j.image.2016.03.002.

[14] S. J. Park, CU encoding depth prediction, early CU splitting termination and fast mode decision for fast HEVC intra-coding, Signal Processing: Image Communication 42 (2016) 79 – 89. doi:http://dx.doi.org/10.1016/j.image.2015.12.006.

[15] J. Xiong, Fast coding unit selection method for high efficiency video coding intra prediction, Optical Engineering 52 (7) (2013) 071504–071504. doi:10.1117/1.OE.52.7.071504.

[16] Y.-C. Lin, J.-C. Lai, Feature-based fast coding unit partition algorithm for high efficiency video coding , Journal of Applied Research and Technology 13 (2) (2015) 205 – 219. doi:http://dx.doi.org/10.1016/j.jart.2015.06.019.

[17] G. Correa, P. Assuncao, L. Agostini, L. da Silva Cruz, Complexity control of high efficiency video encoders for power-constrained devices, Consumer Electronics, IEEE Transactions on 57 (4) (2011) 1866–1874. doi:10.1109/TCE.2011.6131165.

[18] G. Correa, P. Assuncao, L. Agostini, L. Cruz, Coding Tree Depth Estimation for Complexity Reduction

of HEVC, in: Data Compression Conference (DCC), 2013, 2013, pp. 43–52. doi:10.1109/DCC.2013.12.

[19] J. Kim, J. Yang, K. Won, B. Jeon, Early determination of mode decision for HEVC, in: Picture Coding Symposium (PCS), 2012, 2012, pp. 449–452. doi:10.1109/PCS.2012.6213251.

[20] T. Zhao, Z. Wang, S. Kwong, Flexible Mode Selection and Complexity Allocation in High Efficiency Video Coding, Selected Topics in Signal Processing, IEEE Journal of 7 (6) (2013) 1135–1144. doi:10.1109/JSTSP.2013.2271421.

[21] R. H. Gweon, Y.-L. Lee, J. Lim, JCTVC-F045 Early termination of CU encoding to reduce HEVC complexity, Fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) Torino, IT, 2011.

[22] S. Blasi, E. Peixoto, E. Izquierdo, Enhanced inter-prediction using Merge Prediction Transformation in the HEVC codec, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp. 1709–1713. doi:10.1109/ICASSP.2013.6637944.

[23] K. Choi, E. Jang, Early TU decision method for fast video encoding in high efficiency video coding, Electronics Letters 48 (12) (2012) 689–691. doi:10.1049/el.2012.0277.

[24] S.-W. Teng, H.-M. Hang, Y.-F. Chen, Fast mode decision algorithm for Residual Quadtree coding in HEVC, in: Visual Communications and Image Processing (VCIP), 2011 IEEE, 2011, pp. 1–4. doi:10.1109/VCIP.2011.6116062.

[25] L. Shen, Z. Zhang, X. Zhang, P. An, Z. Liu, Fast TU size decision algorithm for HEVC encoders using bayesian theorem detection, Signal Processing: Image Communication 32 (2015) 121 – 128. doi:http://dx.doi.org/10.1016/j.image.2015.01.008.

[26] A. Jimnez-Moreno, E. Martnez-Enrquez, F. D. de Mara, Complexity Control Based on a Fast Coding Unit Decision Method in the HEVC Video Coding Standard, IEEE Transactions on Multimedia 18 (4) (2016) 563–575. doi:10.1109/TMM.2016.2524995.

[27] E. Martinez-Enriquez, A. Jimenez-Moreno, F. Diaz-de Maria, An adaptive algorithm for fast inter mode decision in the H.264/AVC video coding standard, Consumer Electronics, IEEE Transactions on 56 (2) (2010) 826–834. doi:10.1109/TCE.2010.5506008.

[28] A. Jimenez-Moreno, E. Martinez-Enriquez, F. Diaz-de Maria, Mode Decision-Based Algorithm for Complexity Control in H.264/AVC, Multimedia, IEEE Transactions on 15 (5) (2013) 1094–1109. doi:10.1109/TMM.2013.2241414.

[29] HEVC reference software HM13.0. URL http://hevc.hhi.fraunhofer.de/

[30] F. Bossen, JCTVC-F900 Common test conditions and software reference configurations, Fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) Torino, IT, 2011.

[31] F. J. Massey, The Kolmogorov-Smirnov Test for Goodness of Fit, Journal of the American Statistical Association 46 (253) (1951) 68–78.

[32] G.Bjontegaard, Calculation of Average PSNR Differences Between RD-Curves, ITU-T, VCEG-M33, Apr. 2001.