

# Fast Algorithm for Intra Prediction of HEVC Using Adaptive Decision Trees

**Xing Zheng, Yao Zhao, Huihui Bai, Chunyu Lin**  
Institute of Information Science, Beijing Jiaotong University  
Beijing, 100044 - China  
[e-mail: yzhao@bjtu.edu.cn]  
\*Corresponding author: Yao Zhao

*Received February 10, 2016; revised May 15, 2016; accepted May 22, 2016; published July 31, 2016*

---

## **Abstract**

High Efficiency Video Coding (HEVC) Standard, as the latest coding standard, introduces satisfying compression structures with respect to its predecessor Advanced Video Coding (H.264/AVC). The new coding standard can offer improved encoding performance compared with H.264/AVC. However, it also leads to enormous computational complexity that makes it considerably difficult to be implemented in real time application. In this paper, based on machine learning, a fast partitioning method is proposed, which can search for the best splitting structures for Intra-Prediction. In view of the video texture characteristics, we choose the entropy of Gray-Scale Difference Statistics (GDS) and the minimum of Sum of Absolute Transformed Difference (SATD) as two important features, which can make a balance between the computation complexity and classification performance. According to the selected features, adaptive decision trees can be built for the Coding Units (CU) with different size by offline training. Furthermore, by this way, the partition of CUs can be resolved as a binary classification problem. Experimental results have shown that the proposed algorithm can save over 34% encoding time on average, with a negligible Bjontegaard Delta (BD)-rate increase.

---

**Keywords:** fast mode decisions, intra prediction, decision trees, offline training, HEVC

---

This work is supported in part by National Basic Research Program of China (No.2012CB316400), National Natural Science Foundation of China (No. 61210006, 61402034), the Program for Changjiang Scholars, Innovative Research Team in University under Grant IRT201206, Beijing Natural Science Foundation(4154082) and CCF-Tencent Open Fund.

## 1. Introduction

In 2013, High Efficiency Video Coding (HEVC) standard, as the most revolutionary work, has been launched by the joint coding standardization project of ITU-T Video Coding Experts Group (ITU-T VCEG) and ISO/IEC Moving Picture Experts Group (ISO/IEC MPEG) [1]. HEVC can offer an efficient solution to the strong demand of the bandwidth and formats beyond High Definition (HD) resolution, even for the Ultra High Definition (UHD), which have shown more and more popular in video industry.

In HEVC, a picture can be partitioned into many Coding Tree Units (CTU), corresponding with Macro Block (MB) used in previous standards. The CTU is restricted to the square with the size of  $64 \times 64$  pixels. A CTU includes one luma Coding Tree Block (CTB) and two chroma coding tree blocks with corresponding syntax elements. At the same time, through recursive calculation, the CTU can be deeply partitioned into small quad-blocks called Coding Units (CU), whose size are from  $8 \times 8$  to  $64 \times 64$ , as shown in Fig. 1. From this figure, we can see that there are close relations between the maximum allowed CU depth and the encoding complexity. In other words, the greater of the depth, the consumption of encoding time will be longer. Each CU contains more adaptive quadtree structures for the purpose of prediction, so-called Prediction Units (PU), and of transform, so-called Transform Units (TU). Similarly, each Coding Block (CB) can also be split into Prediction Blocks (PB) and Transform Blocks (TB). The main goal of these structures is to adapt the content of the video, so this variable-size standard is particularly suited to large resolution.

The PU is the basic unit used for prediction process in a rectangular shape. One PU can be encoded with one of the modes in candidate sets, which is similar to MB mode of H.264/AVC in spirit. However, the size of PU during the intra prediction can vary from  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$  to  $64 \times 64$ . Each size of PU also contains up to 33 directional prediction modes, one DC prediction mode and one planar mode. Therefore, aiming to the specific size of CU, the encoder has to evaluate the rate-distortion (R-D) cost for 35 times, respectively for the 35 prediction modes. Furthermore, the encoder also searches the optimal partitions of CUs in a recursive manner, which has posed great challenges on real-time applications, especially HD and UHD video formats. In order to reduce the huge complexity for the intra-picture prediction, a fast intra-prediction algorithm has been implemented in the HEVC Test Model (HM). For each of 35 possible prediction modes, a low complexity cost function is always computed using the Sum of Absolute Transformed Differences (SATD) as follows:

$$L_{HAD} = SATD + \lambda_{pred} * Bits_{pred} \quad (1)$$

where  $L_{HAD}$  is the Lagrange cost function,  $\lambda_{pred}$  is the Lagrangian multiplier and  $Bits_{pred}$  stands for the number of bits of the prediction mode. Since  $L_{HAD}$  does not require the implementation of the full encoding and decoding processes, it can speed up the intra prediction to some extent.

The reason why the quad-tree structures [2] are adopted in HEVC is that the encoder can traverse all possible combinations of CU, PU and TU, through RD cost calculation, as with Advanced Video Coding (H.264/AVC), to find the optimal combination for the specific CTU. Therefore, the eventual result of this operation can effectively deal with different

regional characteristics of natural image. For example, in the flat zone, the optimal size of the CU may be 64x64 pixels; however, in a region with complex movements, the optimal size of the CU may be split into 8x8 pixels. While such flexibility leads to more efficient compression, it also increases encoder complexity dramatically.

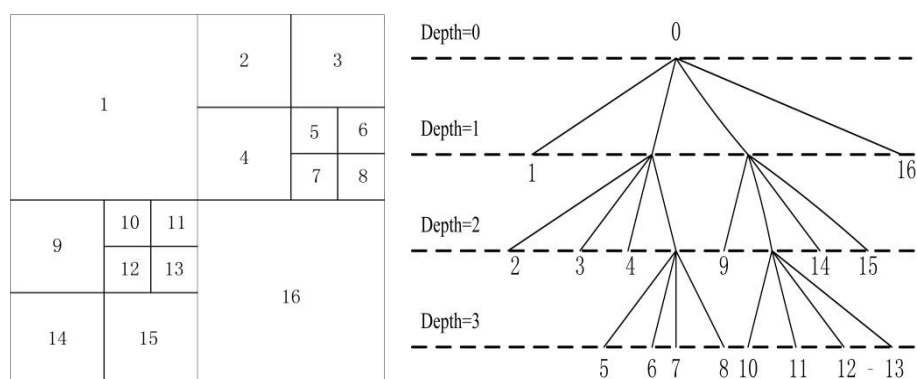


Fig. 1. CU partition based on quad-tree structure

## 2. Related Work

However, with these new tools included in HEVC, consequently, the overall encoding time is larger than before, which is mostly wasted in Rate-Distortion Optimization (RDO) process [3]. Thus, it is necessary to find a novel method to reduce the coding complexity of the intra-frame prediction, a lot of works have been proposed to explore fast algorithms and useful models in the state of encoding for HEVC. Most of the valid algorithms may commit to find the potential links between the CU splitting and CU characteristic. In [4], a fast CU decision is presented using the correlation between the optimal CU depth level and the video content. In [5], Chen has shown a fast intra algorithm based on pixel gradient statistics which employed through the analysis of the video content. In [6], a fast intra-frame prediction algorithm is presented using a Rate-Distortion estimation based on Hadamard Transform (HT). In [7], a fast bottom up pruning algorithm is proposed to reduce the computational cost. In [8], a method based on entropy of CU level is also presented. Generally speaking, the key of these methods is to find out the correlation between the video content and the optimal partition of CU. Through this potential link, we can make precise prediction that aiming to background or flat regions, some specific CU depth levels can be skipped, which can reduce most of the encoding time. In recent years, another popular method which reduces the computational complexity for the encoder is to use parallel computing enabled by many-core processors. In [9], Yan et. al. have proposed a parallel framework to decide the optimal coding unit tree for each image block. Similarly, in [10], to reduce the time of motion estimation (ME) procedure, a parallel framework to decouple ME for different partitions on many-core processors has also been proposed. These type of methods can be more effective for inter prediction compared to intra prediction.

However, few approaches which apply machine learning have been introduced. In [11], Shen proposed a CU size selection algorithm by trying to predict the CU size based on Support Vector Machine (SVM) prediction model. In [12], a fast method aimed to Inter-Prediction is proposed by extracting some typical features related to SKIP mode or Inter 2Nx2N mode. In addition, in [13], we can also learn that machine learning can be used in video transcoding to reduce the encoding time, just as the expected results.

In this paper, to avoid to perform an exhaustive CU size evaluation which used in traditional HM encoder, our approach introduces a data mining classifier generated by machine learning. In order to make a balance of the computation complexity and classification performance, we choose the entropy of Gray-Scale Difference Statistics (GDS) and the minimum of SATD [14] as two important features to design the classifier. For the next stage, these selected features can be applied to decision trees algorithm for the CUs with different size. Through these trees, we can make fairly exact CTU partitioning decisions compared to other works.

This paper is organized as follows. In Section 2, fast CU size decision algorithm based on decision trees is presented. Section 3 shows experimental results and make detailed analysis. The conclusion is summarized in Section 4.

### 3. Proposed Algorithm

In this part, the fast CU partition method will be presented in detail. The task of our method should find a tradeoff between the saving time and rate distortion performance. Therefore, the partition structures used our method should be consistent with the RDO partition structures used in original HM reference software as much as possible. Our algorithm can be primarily divided into three steps. In the first step, fundamentally, the training sample sets should be selected properly, whose coding units can contain a succession of different texture. For example, in these CUs of the sample sets, not only have the smooth areas, but have the regions that possess more movement information. The reasonable training sample sets can greatly improve the accuracy of decision trees for the next step. In the second step, the key of the task is to find two useful features (entropy on account of GDS, the minimum value of SATD after intra prediction) trained by the data mining tool, Waikato Environment for Knowledge Analysis (WEKA) [15]. The selected features should be closely related to decision tree classification. In the final step, by extracting intermediate variable, collected data should be preprocessed and divided into three categories which are corresponding to the CU size of 64x64 pixels, 32x32 pixels and 16x16 pixels, respectively. Consequently, three decision trees will be generated, and the accuracy of these trees can also be measured.

#### 2.1 Training Sample Set

**Table 1.** Training Sequences

Sequence	Frame Rate	Bit Depth	Resolution
<i>BlowingBubbles</i>	50	8	416x240
<i>BQSquare</i>	60	8	416x240
<i>PartyScene</i>	50	8	832x480
<i>BQMall</i>	30	8	832x480
<i>KristenAndSara</i>	60	8	1280x720
<i>FourPeople</i>	60	8	1280x720
<i>BasketballDrive</i>	60	8	1920x1280
<i>Cactus</i>	24	8	1920x1280
<i>Traffic</i>	30	8	2560x1600

Because this positive method is obviously different from original ones, it is noted that the training sample sets should cover a very wide range of content complexity for the region of

CUs as much as possible. In order to achieve this purpose, we have selected the first 30 frames of a collection of different resolution video sequences belonged to JCT-VC test sequences [16] as the training sample sets. In our experiment, nine standard video sequences listed in Table 1 have been used to construct the decision trees of the proposed scheme, because they represent different visual content, motion, and resolution.

## 2.2 Feature Selection

Because this method involves the classification problem, generally speaking, the features used in classification process can have a strong correlation with the partition of CU for the final classifier. Otherwise, the performance of classifier would be unacceptable, what is more, the accuracy of the decision trees will be greatly reduced.

The attributes which can describe CU's content traits are enough, such as edge information, shape type, texture complication and other motion information and so on. However, in consideration of making fast decisions for the split process of CU, we have to consider the computation complexity for a certain kind of feature, so the best deal is to find the balance between the complexity and the well-behaved characteristic for classification problem.

To achieve more precise judgment, a good deal of statistics have already been computed, such as the entropy of CU blocks, the mean of CU blocks, the variance of the CU blocks, the entropy of GDS of CU blocks, and the prediction residual between the original block and the reconstruction block, that is the minimum value of SATD.

Giving attention to both performance and complexity, we observed that the combination of the entropy of GDS and the minimum value of SATD can work better for making a decision whether a CU has to be split or not to be split using the decision trees which have already trained.

### 2.2.1 The Minimum Value of SATD

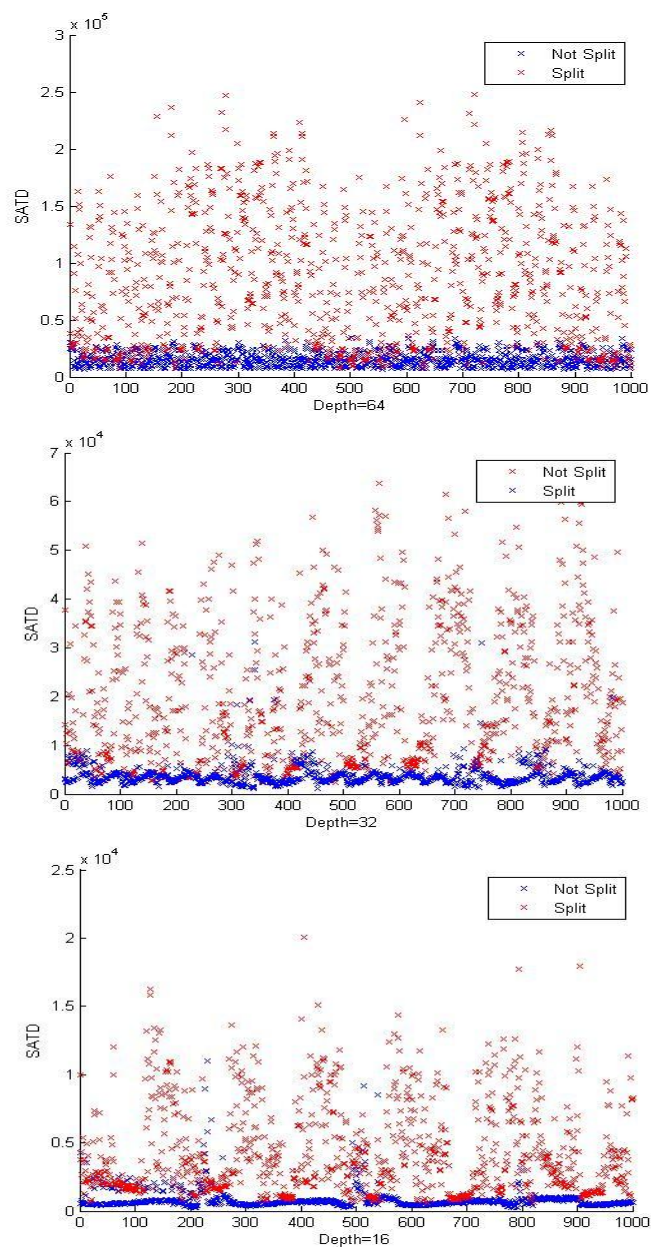
In HEVC, the intra prediction is applied to get rid of the spatial redundancies in the video frame. HEVC provides 35 prediction modes for different size of PUs instead of only 9 prediction modes being available for luma blocks in H.264/AVC. From an increasing number of prediction modes, HEVC can better adapt to the video content especially for the large resolution of video sequences. For each mode, HEVC will always predict the spatial pixels using the neighbor pixels, and through this operation, the reconstruction block will be presented for us. What is more, the metric of this prediction method is co-called SATD which served as the valid feature in our experiment. To calculate the cost of particular mode, the value of SATD is used where transform is Hadamard Transform. The formula for calculation is as follows:

$$\text{Sum of Absolute Difference (SAD)} = \sum_{i,j} |s_A(i,j) - s_B(i,j)| \quad (2)$$

$$\text{Hadamard Transformed SAD (SATD)} = \sum_{i,j} |HT(i,j)| \quad (3)$$

where  $s_A(i,j)$  and  $s_B(i,j)$  denote the  $(i,j)^{th}$  sample in blocks  $A$  and  $B$  of the same size, respectively.  $HT(i,j)$  in (2) is the  $(i,j)^{th}$  coefficient of a block that is obtained by applying Hadamard transform to the block difference between blocks  $A$  and  $B$ . By traversing the residual matrix, the sum of the absolute value of each element can be

computed, which is used to the value of SATD. For a flat region, the accuracy of prediction can be higher, so the value of SATD can be smaller. On the contrary, for the area that contains complex motion information, the value of SATD seems to be a little larger. Therefore, to some extent, the values of the SATD can better reflect the video content complexity. The more important reason why we choose the SATD is that the minimum value of SATD can be gained easily in HM reference software. Taking the analysis into consideration, we choose the value of SATD as the input property for the decision trees.



**Fig. 2.** The CU's SATD values at different depth



To further confirm the feature of SATD values have a close relationship with the split process of CU optimal depth level, we encode the first 20 frames of the standard test video sequence *BQTerrace* and recode the minimum value of SATD in HM 10.1. These SATD values will be split into two classes. One class contains the data for the split of CUs and the other class contains the data for the non-split of CUs. Each of class also includes three types of data, corresponding to the CU size of 64x64, 32x32 and 16x16. We select one thousand CU blocks that will be split and the same number of CU blocks that will be non-split. The distribution of these values are shown in Fig. 2, where the blue dots on behalf of the CUs that will be split, to the contrary, the red dots on behalf of the CUs that won't be split further.

From the above figures, we can see clearly that the CUs corresponding with the smaller values of SATD have the smaller possibility to be split, inversely, those CUs with the bigger values of SATD have the higher possibility chance to be split. Therefore, these experiments are conducted to demonstrate the effectiveness of the value of SATD which is helpful to the training of the decision trees.

### 2.2.2 The Entropy of GDS

In the proposed algorithm, the entropy based on GDS plays an important role in the classification problem. The GDS method [17] is suggested in an attempt to define texture measures correlated with human perception. The gray-tone differences for each pixel can be calculated as follows:

$$g_{\Delta}(x, y) = g(x, y) - g(x + \Delta x, y + \Delta y) \quad (4)$$

where  $g(x, y)$  is pixel values at the point of  $(x, y)$  aiming to the specific range of the CUs, and  $\Delta x$  represents the offset value relative to the horizontal position of the  $x$ , similarly,  $\Delta y$  represents the offset value relative to the vertical position of the  $y$ , in the process of our statistical results shows that the performance is better when the  $\Delta x=1$  and  $\Delta y=1$ , so the values of  $g_{\Delta}(x, y)$  represents the differences of the pixels.

It is assumed that there are  $M$  possible values of  $g_{\Delta}(x, y)$  so when we can change the values of  $x$  and  $y$  in the whole CU region, the frequency of a particular value for  $g_{\Delta}(x, y)$  can be stored. Through these stored data, we can draw two-dimension histogram of the value of  $g_{\Delta}(x, y)$ . Then we can obtain the corresponding probability while  $g_{\Delta}(x, y)$  is assigned different values. Different useful parameters of image features can be worked out from the histogram, which can be used to quantitatively describe the first-order statistical properties of the CU region. A large number of features can be calculated using the GDS method for the purpose of texture discrimination, such as the contrast, second-order moments, entropy and mean. In view of low computation complexity, we choose the entropy as the second feature for the training of decision trees.

It is well known that the entropy is a very effective measure to estimate the complexity of the video texture. If the region of the CU is more smooth, the value of the entropy can be smaller; otherwise, if the region of the CU is more complicated, then the value of the entropy can be larger. The equation for the entropy is expressed as follows:

$$H(x) = -\sum_{i=0}^{i=j} p_i \log_2 p_i \quad (5)$$

where  $H(x)$  is the entropy corresponding to the CU region,  $p_i$  presents the probability of the element  $i$ , and  $j$  is the number of elements. By iterating through all the possible pixels in the CUs, the final  $H(x)$  can be calculated and stored as an attribute of an instances used in decision trees in the training phrase.

The Fig. 3 shows the precision of the split for different size of CUs using the entropy of GDS and the conventional entropy of pixel-levels.

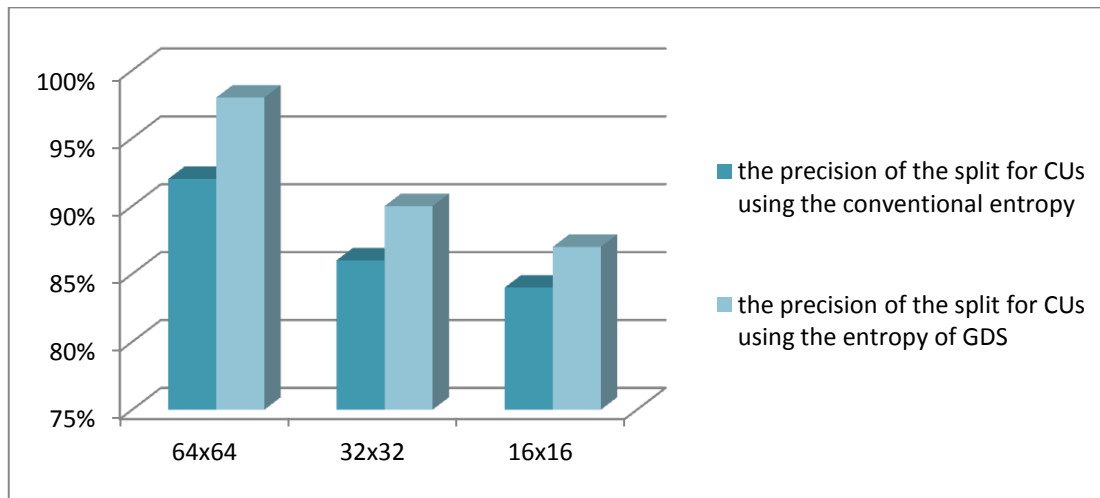


Fig. 3. The precision of the splitting for different CU size using two types of entropy

From Fig. 3, we can see that the methods using different kinds of entropy have different performance of classification for the split process of the CUs. The method using the entropy of GDS has shown the better performance of classification for the different size of CUs, so the method will be adopted in our algorithm.

### 2.3 Decision Tree

In this paper, the tool applied to guide the data mining process is the WEKA with 3.6.12 version. The machine learning contains many approaches, and most of them have already been realized in WEKA. The file format for WEKA is an Attribute-Relation File Format (ARFF). In the specific case of building decision trees, the last line of this format file can identify the class attribute, which in our experiment indicates that whether the CU can be split or not be split.

Furthermore, we choose C4.5 classifier in WEKA for its good performance. For C4.5 classifier, the input is the ARFF file and the output is the well-trained decision trees. When building the decision trees using the C4.5 algorithm for the CU early termination, the importance of each attribute can be evaluated through the Information Gain Attribute Evaluation (IGAE) which be used to classify the data into the different classes in WEKA. This indicator will employ the Kullback–Leibler divergence (KLD) as the only metric to choose the most valuable attribute. So, the information gain of a feature shows that how important it is for the process of training a decision tree aiming to the different size of CUs.

As shown in Fig. 4, the decision tree contains two parts, nodes and arcs. The nodes represent tests performed on the attributes and the arcs are the prediction results of certain tests. In our scheme, the combination of two features can be seen as an instance which stands



for the block of CUs. The C.4.5 algorithm will take all instances as the inputs based on the values of KLD, and attain the thresholds in the current stage for classification. In Fig. 4, a simple example has shown to explain the split process of the decision trees. For a given CU block, the values of SATD and GDS entropy of this block can be calculated and then the path can be traced from the root node to the leaf node. If the output of the leaf node is 1, which represents split decision for the current CU block; otherwise, the current CU block will not be split.

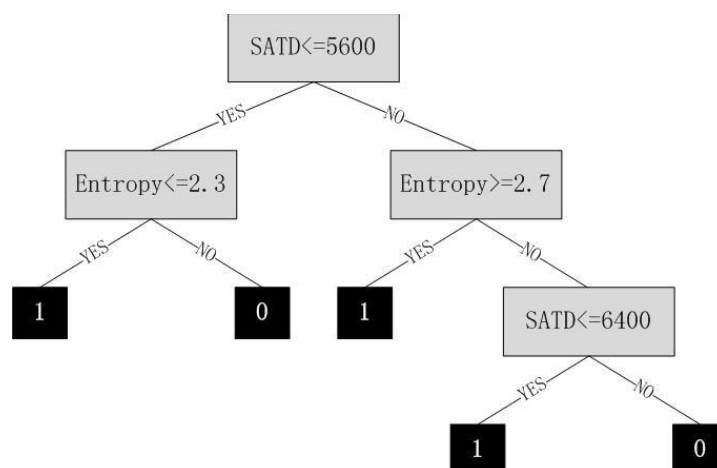


Fig. 4. The splitting process of the decision trees

Based on the above analysis, the characteristics of the well-trained decision trees can be shown in Table 2. In this table, the accuracy for the different size of CUs along with their depth, the number of the leaf nodes have already be presented for us. We can see that the decision trees can obtain the better decision accuracy with the percentage of the value can reach more than 80%. At the same time, because the values of depth for the decision trees is lower, so the time attached to the encoder can be accepted.

Table 2. The structures of the decision trees for training sequences in Table 1

CU Size	Decision Accuracy	Depth	Leaf Node
64x64	83.6%	6	22
32x32	81.7%	8	28
16x16	80.2%	8	36

## 4. Simulation Results

In the proposed algorithm, the training data sets include a half of CUs that will be split into small CUs and the other half of CUs that will not be split into small CUs, which can reduce the problem of the classification imbalance. We can get the related training data for the features about the minimum value of SATD and the entropy of the gray difference statistics, through collecting the intermediate variables during the encoding procedure of the nine video sequence sets in Table 1, using four quantization parameters (QP) values (22,27,32,37). These data will be used to train the decision trees for prediction.

Furthermore, the rest of the nine video sequences sets, as shown in **Table 3**, can be used to validate the performance of the decision trees. What calls for special attention is that the nine sequences used in the training stage should not be included in the testing stage. The metric of the proposed algorithm is measured by *PSNR* difference  $\Delta PSNR$ , bit rate difference  $\Delta Bitrate$  and time saving  $\Delta T$ . We encode the up to 100 frames for each test sequences in **Table 3**. The experiment condition is set up as “All Intra-Main” (All-Main) configuration based on HM10.1.

**Table 3.** Testing Sequences

Sequence	Frame Rate	Bit	Resolution
<i>BasketballPass</i>	50	8	416x240
<i>RaceHorses</i>	30	8	416x240
<i>BasketballDrill</i>	50	8	832x480
<i>RaceHorses</i>	60	8	832x480
<i>Johnny</i>	60	8	1280x720
<i>BQTerrace</i>	50	8	1920x1280
<i>Kimono</i>	24	8	1920x1280
<i>ParkScene</i>	50	8	1920x1280
<i>PeopleOnStreet</i>	30	8	2560x1600

**Table 4.** Performance comparison between Huang’s and the proposed algorithm

Sequence		Proposed		Huang’s	
		BD -Rate (%)	$\overline{\Delta T}$ (%)	BD -Rate (%)	$\overline{\Delta T}$ (%)
Class A 2560x1680	<i>PeopleOnStreet</i>	1.71	29.95	0.40	19.31
Class B 1920x1080	<i>Kimono</i> <i>BQTerrace</i> <i>ParkScene</i>	1.77	56.53	0.79	7.25
		2.18	30.52	0.32	20.39
		1.71	33.78	0.73	19.37
Class C 832x480	<i>RaceHorses</i> <i>BasketballDrill</i>	1.90	30.64	0.38	23.19
		1.52	22.25	0.43	24.44
Class D 416x240	<i>BasketballPass</i> <i>RaceHorses</i>	2.08	32.64	0.37	18.70
		0.27	14.20	0.20	24.53
Class E 1280x720	<i>Johnny</i>	4.83	59.58	0.89	14.43
Average		1.994	34.45	0.501	19.06

**Table 4** shows the final results of our algorithm with  $\Delta Bitrate$ ,  $\Delta PSNR$  and  $\Delta T$ . Furthermore, affected by different QPs, the mean of  $\Delta T$  (noted by  $\overline{\Delta T}$ ) takes the place of  $\Delta T$ , which can be calculated by the following equation.

$$\overline{\Delta T} = \frac{1}{4} \sum \Delta T \times 100\% \quad (6)$$

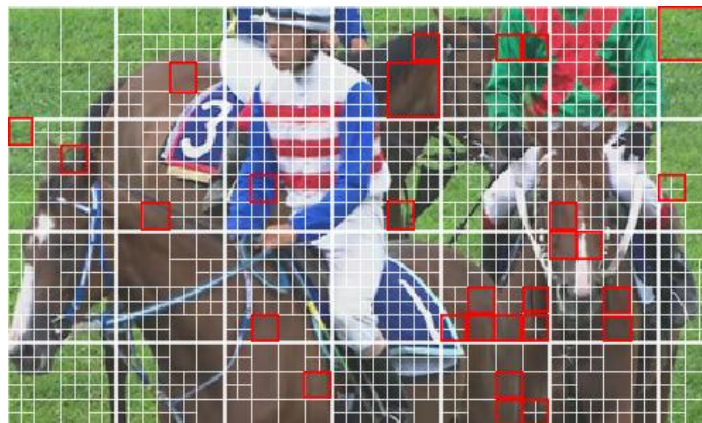
with

$$\Delta T = \frac{Time_{HM10.1} - Time_{Proposed}}{Time_{HM10.1}} \quad (7)$$

In **Table 4**, we can see that the proposed fast CU partition algorithm can achieve about 34.45% reduction of the total encoding complexity, and the Bjontegaard delta (BD) rate exhibit 1.994% increment on average. Moreover, compared with Huang's method in [7], the proposed algorithm can save more encoding time with a tolerable bitrate increase.

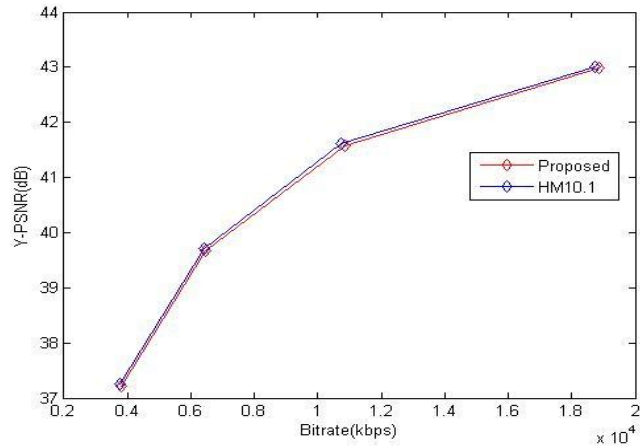


(a)

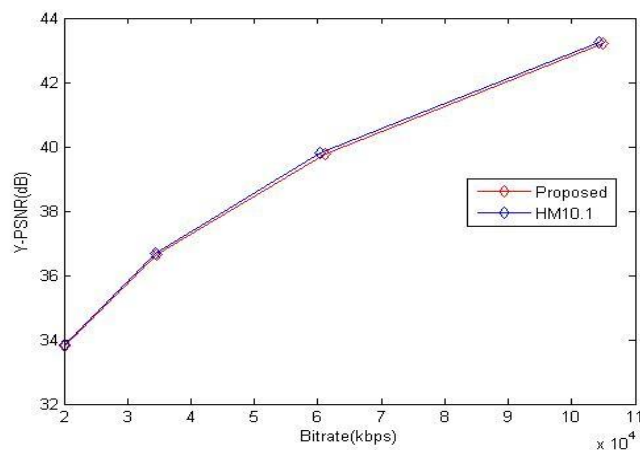


(b)

**Fig. 5.** The CU Partition Structures for RaceHorses (a) The RDO algorithm in HM10.1; (b) Proposed algorithm



(a) Kimono



(b) PeopleOnStreet

**Fig. 6.** RD Curves of HM10.1 and our proposed algorithm

**Fig. 5** shows the partition comparison between the proposed and anchor RDO algorithm in HM10.1 for the sequences RaceHorses. In the figures, white lines can be used for the same partition while red lines for different partition. From **Fig. 5**, we can observe that most of the CUs with 64x64 pixels have almost been split, even if in some flat regions, in other words, the output of the decision tree of CU with 64x64 pixels is relatively accurate. The similar performance of the decision trees appears in CUs with size of 32x32 pixels and 16x16 pixels.

In addition, **Fig. 6** has displayed the rate distortion performance of the proposed algorithm and RDO algorithm in HEVC for the specific sequence Kimono and PeopleOnStreet. We can clearly see that the proposed algorithm can maintain the rate distortion performance for Y component compared with the original RDO algorithm in HEVC.

## 5. Conclusion

We present a fast CU partitioning approach for HEVC Intra-Prediction using machine learning. Considering the balance between the performance and complexity, entropy of the GDS and the minimum value of SATD are selected as features of the CU blocks. According to these selected features, three decision trees can be constructed to predict the early termination of CU partition process.. The experimental results indicated that the proposed algorithm can achieve over 34% encoding time reducing on average with negligible coding efficiency loss.

## References

- [1] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 10 (JCTVC-L1003)," in *Proc. of JCT-VC Meeting (Joint Collaborative Team of ISO/IEC MPEG & ITU-T VCEG)*, Geneva, Switzerland, January 14-23, 2013.
- [2] W. J. Han, J. Min, I. K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M. S. Cheon, N. Shlyakhov, K. McCann, T. Davies and J. H. Park, "Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1709-1720, December, 2010. [Article \(CrossRef Link\)](#).
- [3] X. Li, M. Wien, and J. R. Ohm, "Rate-complexity distortion optimization for hybrid video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 7, pp. 957-970, July, 2011. [Article \(CrossRef Link\)](#).
- [4] L. Shen, Z. Zhang and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 207-213, February, 2013. [Article \(CrossRef Link\)](#).
- [5] G. Chen, Z. Pei, L. Sun, Z. Liu and T. Ikenaga, "Fast intra prediction for HEVC based on pixel gradient statistics and mode refinement," in *Proc. of IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, pp. 514-517, July 6-10, 2013. [Article \(CrossRef Link\)](#).
- [6] Y. Kim, D. Jun, S. H. Jung, J. S. Choi and J. 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, April, 2013. [Article \(CrossRef Link\)](#).
- [7] H. Huang, Y. Zhao, C. Lin, and H. Bai, "Fast bottom-up pruning for HEVC intra frame coding," in *Proc. of Visual Communications and Image Processing (VCIP)*, pp. 1-5, November 17-20, 2013. [Article \(CrossRef Link\)](#).
- [8] M. Zhang, J. Qu, and H. Bai, "Entropy-Based Fast Largest Coding Unit Partition Algorithm in High-Efficiency Video Coding," *Entropy*, vol. 15, no. 6, pp. 2277-2287, June, 2013. [Article \(CrossRef Link\)](#).
- [9] C. Yan, Y. Zhang, J. Xu, F. Dai, L. Li, Q. Dai and F. Wu, "A Highly Parallel Framework for HEVC Coding Unit Partitioning Tree Decision on Many-core Processors," *IEEE Signal Processing Letters*, vol. 21, no. 5, pp. 573-576, May, 2014. [Article \(CrossRef Link\)](#).
- [10] C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai and F. Wu, "Efficient Parallel Framework for HEVC Motion Estimation on Many-Core Processors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2077-2089, December, 2014. [Article \(CrossRef Link\)](#).
- [11] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 1-11, January, 2013. [Article \(CrossRef Link\)](#).
- [12] G. Correa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC Encoding Decisions Using Data Mining," *IEEE Transactions on Circuits and Systems for Video*

- Technology*, vol. 25, no. 4, pp. 660-673, April, 2015. [Article \(CrossRef Link\)](#).
- [13] G. Fernandez-Escribano, J. Bialkowski, J. Gamez, H. Kalva, P. Cuenca, L. Orozco-Barbosa and A. Kaup, "Low-Complexity Heterogeneous Video Transcoding Using Data Mining," *IEEE Transactions on Multimedia*, vol. 10, no. 2, pp. 286-299, February, 2008. [Article \(CrossRef Link\)](#).
- [14] V. Sze, M. Budagavi and G. J. Sullivan, *High Efficiency Video Coding (HEVC) Algorithm and Architectures*, 1st Edition, Springer International Publishing, Switzerland, 2014.
- [15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, June, 2009. [Article \(CrossRef Link\)](#).
- [16] JCT-VC test sequences. [Online] <ftp://hevc@ftp.tnt.uni-hannover.de/testsequences/>
- [17] D. Li, Y. Chen, *Computer and Computing Technologies in Agriculture VIII*, 1st Edition, Springer International Publishing, China, 2015. [Article \(CrossRef Link\)](#).



**Xing Zheng** received her B.S. degree from ShanXi University, China, in 2014. He is currently a Master student in Beijing Jiaotong University, China. His research interests are fast encoding of HEVC.



**Yao Zhao** received the BS degree from Fuzhou University, China, in 1989, and the ME degree from Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the PhD degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), China, in 1996. He became an associate professor at BJTU in 1998 and became a professor in 2001. From 2001 to 2002, he was a senior research fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is currently the director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He serves on the editorial boards of several international journals, including as associate editors of *IEEE Transactions on Cybernetics*, *IEEE Signal Processing Letters*, and an area editor of *Signal Processing: Image Communication* (Elsevier), etc. He was named a distinguished young scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013. He is a senior member of the IEEE.





**Huihui Bai** received her B.S. degree from Beijing Jiaotong University, China, in 2001, and her Ph.D. degree from Beijing Jiaotong University, China, in 2008. She is currently a professor in Beijing Jiaotong University. She has been engaged in R&D work in video coding technologies and standards, such as HEVC, 3D video compression, multiple description video coding (MDC), and distributed video coding (DVC).



**Chunyu Lin** was born in Liaoning Province, China. He received the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2011. From 2011 to 2012, he was a Postdoctoral Researcher with the Multimedia Laboratory, Ghent University, Ghent, Belgium. He is currently an Associate Professor with Beijing Jiaotong University. His current research interests include image/video compression and robust transmission, 2D-to-3D conversion, and 3-D video processing.