IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 5, 2020, accepted February 25, 2020, date of publication March 2, 2020, date of current version March 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977422

Hierarchical Classification for Complexity Reduction in HEVC Inter Coding

YU LU[1,2], (Member, IEEE), XUDONG HUANG[1], HUAPING LIU[2], (Senior Member, IEEE), YANG ZHOU[1], HAIBING YIN[1], AND LIQUAN SHEN[3]

[1]School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China
[2]School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331, USA
[3]School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

Corresponding author: Yu Lu (luy4@oregonstate.edu)

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY18F010015, in part by the National Natural Science Foundation of China under Grant 61671282 and Grant 61972123, and in part by the China Scholarship Council.

ABSTRACT In order to adapt to various real-time applications, fast coding algorithms for high efficiency video coding (HEVC) standard maintain a hot research topic in recent years. In this paper, a complexity reduction algorithm based on hierarchical classification for HEVC inter coding is proposed. It consists of five fast algorithms which is accomplished by hierarchical classification trees at coding unit (CU) level, prediction unit (PU) level and transformation unit (TU) level respectively. At the beginning of proposed algorithm, intra features and inter features which describe the texture and context properties of CU, PU and TU are extracted from the training set. Then the classification trees for CU, PU and TU are generated by carefully selecting features and designing the classification criteria. By analyzing the spatiotemporal correlation, two strategies including early termination and early split are applied to fast coding by referring to these classification trees. The objective evaluation demonstrates that the proposed algorithm can significantly reduce coding complexity with little compression loss. Particularly the subjective evaluation based on similarity measurement for color histogram approves that decoded video quality between the original HM16.9 algorithm and the proposed algorithm is nearly identical.

INDEX TERMS HEVC, inter coding, fast coding, coding unit, prediction unit, transformation unit, classification tree, histogram similarity.

I. INTRODUCTION

With the development of high definition videos such as digital broadcasting and mobile video, video services are quickly evolving with the popularity of the internet and mobile networks. But it requires more bandwidth to accommodate higher data stream. Although computer storage capacity and network bandwidth are increasing, the previous H.264/AVC video coding standard will lead to a large amount of data redundancy in encoding the high-resolution video. In order to solve this problem, the high efficiency video coding (HEVC) standard is developed by the Joint Collaborate Team on Video Coding (JCT-VC) which consists of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG)[1]. It can provide higher compression efficiency and more flexible network adaptability. HEVC achieves 50% bit-rate coding gains while maintains the similar decoding quality in comparison with preceding H.264/AVC standard. In order to adapt to various video content, HEVC adopts hybrid coding structure with many advanced compression technologies, including quadtree-based data partition, intra prediction, inter prediction, multi-reference frame motion estimation, sample adaptive offset (SAO) filter and so on. In order to reduce the spatiotemporal redundancy, coding redundancy and visual redundancy in the high definition video, the coding tree unit (CTU) is coded as an independent processing unit in HEVC. A CTU can be encoded into diverse blocks to accommodate different video content. Each coding unit (CU) at each depth can be encoded with different prediction unit (PU) and different transformation unit (TU). And the rate distortion (RD) costs of all possible combinations of CU, PU and TU are calculated by the method of exhaustive search. Then the optimal combination of coding pattern is selected in terms of

The associate editor coordinating the review of this manuscript and approving it for publication was Nilanjan Dey.

the minimum RD cost. However, these strategies have greatly increased the coding complexity, which make it difficult in real time applications.

In order to tackle this problem, a large amount of fast coding algorithms have been proposed. Most of them are based on the statistical analysis of block partition and prediction mode selection. In these algorithms, various features such as texture features and coding features are firstly extracted during video coding. Then the correlation among these features and optimal CU depth, PU mode and TU depth is exploited. Based on it, early decision algorithms for block partition and mode selection are fulfilled. For example, a fast coding algorithm by early mode selection based on the intra block similarity is presented in [2]. And in [3], the grayscale similarity and inter-view correlation are jointly applied to early decide the PU mode. An early termination of CU partition algorithm is proposed in [4]. At first, the feature of CU depth history is recorded. Then its average value and standard deviation are calculated and used to define the CU depth range. Consequently, the trivial CU depth is skipped. In [5], according to the correlation between edge feature and CU depth, the optimal CU depth can be predicted. In [6], the spatial and temporal homogeneity is used to classify different size of CU so that the CU quad-tree division process can be early stopped. And in [7], the depth correlation coefficient is computed in terms of scene content change. It can be used to predict CU depth range in the homogeneous region and the heterogeneous region. In [8], motion consistency between current frame and its adjacent frame is used to classify the motion region. Then based on the statistical RD model, the PU mode can be early decided. In addition, a fast mode decision algorithm is proposed in [9], where adaptive ordering of modes is used to skip the unnecessary modes according to RD cost and bit cost. Traditionally CTU is visited from top to bottom at each level of recursion to select the best depth. But a conversely bottom to top visiting order for CTU partition is proposed in [10] where the coding information acquired from the small CU is used to reduce testing options in the large CU. Specially in [11], targeting the TU coding, a fast TU split algorithm based on the Bayesian decision rule by employing the correlations between current TU and its adjacent TUs is proposed. In [12], in order to decrease the computational complexity in motion estimation, an adaptive algorithm for skipping fractional-pixel motion estimation is presented to fast select the reference frame based on content similarity between the parent PU and the child PU. In [13], combining the spatiotemporal and depth correlation with the classification of motion activity, a fast PU decision algorithm is addressed. In [14], CU is classified into motion or static block at first. Then for motion block, CU is split by the texture quad-tree model and temporal correlations model of CU. While for the static block, the depth range and prediction mode of largest CTU are predicted. Additionally, an early termination of reference frame selection method together with early decision of SKIP mode method is proposed in [15]. In our previous work [16], early decision for CU depth and PU mode is implemented by fully exploiting the correlated information among luminance, gradient and neighboring blocks in screen video coding. Also a fast inter-mode decision algorithm by jointly using the correlations of PU, motion vector and RD cost at different CU depth is proposed in [17]. Recently some fast coding algorithms involved with machine learning are published. In these algorithms, block partition and mode selection in video coding are modeled as data classification via online learning or offline learning. In [18], an efficient CU decision method based on machine learning for flexible complexity allocation is proposed. It mainly consists of the three-output classifier which is designed to control the risk of false prediction, and a sophisticated RD complexity model is derived for the optimal training parameter determination. In [19], a fast CU decision algorithm is addressed by utilizing the Fisher's linear discriminant analysis and the k-nearest neighbor classifier. The statistical data used in it are updated by an adaptively online learning method. In order to optimize the traversal process of coding trees, prediction units and residual quadtrees, a fast coding algorithm based on decision tree using the C4.5 algorithm is presented in [20]. In [21], an early CU termination classifier relying on the framework of reinforcement learning is proposed, where the RD cost comparison process is modeled as the Markov decision process. In addition in [22], the dimension reduction and classification techniques including stepwise regression, random forest, variable selection, principal component analysis, polynomial classifiers are employed for fast CU split. On the other side, the support vector machine (SVM) based fast coding algorithm is reported in [23] and [24]. In [23], the CU decision and PU selection are determined by binary and multi-class SVM algorithm. And the performance is further improved by a learning scheme based on a multiple-reviewer system and a flexible complexity allocation algorithm. Again in [24], a fuzzy SVM for fast CU decision algorithm based on RD cost optimization is proposed, where CU partition is regarded as a cascaded process of multi-level classification. In our previous work [25], CU depth is early decided by the CU partition threshold which is acquired by online learning the random sample set. One merit of machine learning based algorithm is that coding features can be acquired by adequately exploiting the massive coding data by the learning tool, which is more beneficial to fast coding. Consequently, it motivates us to propose a new algorithm.

In this paper, a fast inter coding algorithm is proposed based on hierarchical classification trees generated by the classification and regression tree (CART) algorithm [26]. The decision for CU, PU and TU is modeled as the process of multiple-level classification. There are several characteristics in the proposed algorithm. At first, new classification features for CU, PU and TU based on statistical analysis are exploited. Then classification trees are generated by carefully selecting features and designing the classification criteria. In addition, different from most of current algorithms using early termination only, both early termination and early split are applied in this paper. Besides the objective evaluation, the histogram

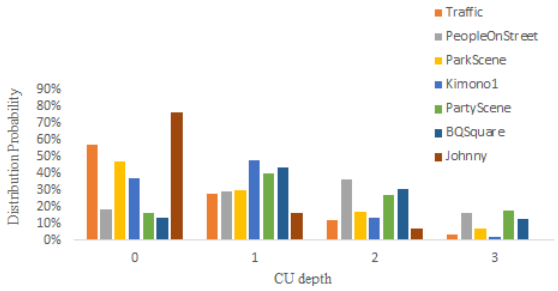

FIGURE 1. Distribution probability of CU depth.

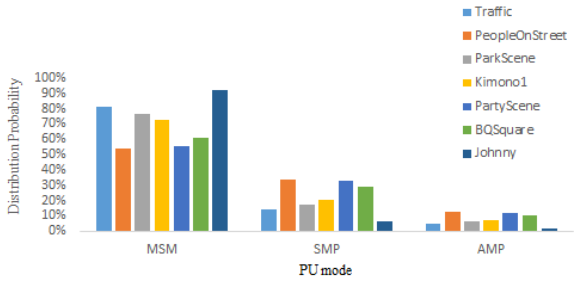

FIGURE 2. Distribution probability of PU mode.

similarity is employed to measure the subjective video quality. The structure of this paper is arranged as several sections. Section II explains the primary theory of HEVC coding, and the specific description about the proposed algorithm is presented in section III. The experimental results are given in section IV and the conclusion is drawn in section V.

II. PRIMARY THEORY

Video compression is a block-based coding framework involved with hybrid prediction and transformation modules, where prediction coding, transformation coding, and entropy coding are applied. At first, video frame is divided into non-overlapped coding blocks. In order to remove spatiotemporal redundancy, intra prediction and inter prediction are used to obtain the predicted value of each pixel. Then it is subtracted from the original value to acquire the residual value. Afterwards DCT transformation and quantization are used to attain the quantized transformation coefficients which are finally applied to entropy encoding. Meanwhile, the obtained transformation coefficients are inversely transformed and quantized. Then the result is filtered and stored in the reference frame list as inter prediction for the subsequent frames.

A. CODING STRUCTURE OF CU, PU, and TU

Three flexible coding structures including CU, PU and TU are used in HEVC standard to improve the compression efficiency. Based on the CU structure defined in the HEVC standard, each frame begins coding from a largest coding unit (LCU) with size of 64 × 64, which is recursively divided into four sub-CUs in terms of the quad-tree structure until the smallest coding unit(SCU) with size of 8 × 8. There are four depths of CU from 0 to 3 corresponding to four sizes of CU from 64 × 64 to 8 × 8. Meanwhile each CU coding is associated with the decision of PU and TU. The distribution probability of CU depth for seven different sequences which are encoded under the configuration of low delay P (LDP) is shown in Fig. 1. It can be seen that CU depth distribution in some sequences such as Johnny hold high probability in the small depth and low probability in the large depth. Therefore, early termination of CU partition may reduce the complexity of CU decision under this condition. On the other hand, sequence BQSquare has the reverse probability distribution that depth 0 is less probably selected. At this case, CU coding can be speeded up if early split of CU is applied at depth 0. Therefore, fast coding strategies of early termination and early split are both employed in this paper.

At each CU depth level, PU is the basic unit for intra/inter prediction which contains vital prediction information. PU mode includes MERGE/SKIP mode (MSM), symmetric-mode-prediction (SMP) mode and asymmetric-mode- prediction (AMP) mode. The SMP mode includes four PU modes of 2N×2N, N×2N, 2N×N, N×N and the AMP mode includes four PU modes of 2N×nU, 2N×nD, nL×2N, nR×2N. The MSM mode is only applied to the 2N×2N mode and its motion information such as motion vector is expressed as motion index which is obtained by exploiting the temporal and spatial correlation between the adjacent blocks. At the decoder, the whole motion information can be recovered from the motion index. Therefore, plenty of coding bits can be saved. In addition, the SKIP mode is a special MSM mode where only a skip tag is transmitted. When it is adopted, high compression rate is achieved. Seen from Fig. 2, MSM mode occupies the maximum distribution probability in all modes. Therefore, MSM mode is the most preferred PU mode because a large number of time and coding bits can be saved. It is usually chosen as the optimal prediction mode for the static or motion-smooth regions.

Similar to CU split, TU partition is implemented by using the residual quad-tree (RQT). The TU size can be adaptively determined from 4 × 4 to 32 × 32 according to the local characteristics of prediction residual. It is a basic unit for transformation and quantization of prediction residual block. And TU size cannot exceed the corresponding CU size. Subject to each CU size, TU can be divided into different size. Associated to depth 0 of CU, TU can be split into size of 32 × 32 or 16 × 16. For the depth 1 of CU, TU can be split into size of 32 × 32 or 16 × 16 or 8 × 8. In terms of depth 2 of CU, TU can be split into size of 16 × 16 or 8 × 8 or 4 × 4. And TU can be split into size 8 × 8 or 4 × 4 when the depth of CU is 3. The small TU is suitable for the high frequency region where the texture is complicated and the luminance changes discontinuously. While the large TU is suitable for the low frequency region with slow variation.

B. CODING OPTIMIZATION BASED ON RD COST

In HEVC standard, the selection of best mode for CU, PU and TU is depended on the RD cost. At first, the RD costs of all

possible modes are calculated then the coding mode with the least RD cost is regarded as the optimal pattern. This strategy can adapt to various video contents including static background and moving objects. Based on this mechanism, each coding block can find its consistent prediction block in the reference frame during inter coding. If the resulting residual is small, the number of coding bits will greatly decrease. The Lagrangian rate-distortion-optimization (RDO) function [27] is used for the optimal mode decision, which is formulized as,

$$L_i = D_i(rec, org) + \lambda \cdot R_i \tag{1}$$

where L_i is the RD cost of mode i, $D_i(rec, org)$ represents the distortion of the reconstructed value in comparison with the original value. And R_i represents the number of bits required to encode the current mode information. The parameter λ is related to the Quantization Parameter (QP), which is used to control the balance between $D_i(rec, org)$ and R_i. Since there is a close relationship between RD cost and distortion in (1), it is considered as the important coding features in the proposed algorithm.

After CU is split at each depth, four sub-CUs are obtained. The division process continues until the depth of current CU reaches 3. Then starting from SCU, the sum of RD costs of four sub-CUs is compared with the RD cost of their parent CU at the previous depth. The CU depth with minimum RD cost is retained. Then the next similar comparison continues until the comparison with the RD cost of LCU is completed. Finally the optimal CU depth is decided by the minimum RD cost. General speaking, large CU is coded for the flat region with simple texture in order to save coding bits. While for the complex areas, the small CU coding is usually adopted to describe local details. Thus prediction accuracy and coding efficiency can reach a balance. After PU mode selection, the residual for each CU is obtained by subtracting the original block from the motion compensated block. Small residual indicates current CU will not be divided into the smaller size because of good matching between the original block and prediction block. At this case, it can attain good decoding quality but suffer the increase of bit rates. On the contrary, large residual implies low prediction accuracy, which results in the degradation of decoding quality and the decrease of bit rates. In this situation, the RD cost is high so that current CU will continue to be split into the smaller CU.

The inter mode selection is performed at each CU level. At first, the MSM mode prediction is performed and followed by the SMP mode prediction and the AMP mode prediction. It is noted that the AMP mode prediction is enabled only when the CU depth is less than 3. The RD cost for each PU mode is calculated and compared, and the prediction mode with the minimum RD cost is selected as the best mode. In addition, the MSM mode is preferred because it can reduce the computational workload by sharing the inter motion information with the adjacent block. In Fig. 2, it also exhibits that the MSM mode accounts for the highest proportion. Consequently, if the MSM mode can be decided in advance, the remaining PU mode decision will be skipped.

Similar to CU split, TU partition is performed based on the RQT structure. After the residual block is coded, its energy usually concentrates on the low frequency region. If the energy of TU is scattered, many transformation coefficients will extend to the high frequency region. Under this condition, the current TU will be divided into four smaller TUs. Large TU can favorably condense the energy while small TU can maintain more details in the video. And large TU can attain satisfactory quality in the smooth area but cause visual distortion at the edges of the object. Comparatively small TU can alleviate the edge distortion but aggravate the coding complexity. In order to attain the optimal TU depth, a traversal RD optimization algorithm depending on RQT structure is performed in the HEVC coding, which results in heavy workload.

In summary, the flexible block partition and mode selection are beneficial to video compression but introduce high coding complexity. Obviously fully traversing all block depth and prediction mode causes significant increase in the coding complexity for HEVC. If the decision for CU, PU and TU can be early terminated or early split, the corresponding calculation of RD cost will be exempted. Such that the inter coding can be accelerated.

III. THE PROPOSED ALGORITHM

There are several main steps in the proposed algorithm. At first, the selected frames from seven sequences form the training set. Then it is coded by the normal HEVC algorithm to extract intra features and inter features which are applied to generate the classification trees. Afterwards they are used to implement fast coding algorithm consist of early termination of CU partition based on temporal similarity (ETCU_T) algorithm, early split of CU partition (ESCU) algorithm, early termination of PU selection (ETPU) algorithm, early termination of TU partition (ETTU) algorithm and early termination of CU partition based on spatial similarity (ETCU_S) algorithm.

A. TRAINING SET

In order to obtain different type of samples, seven sequences representing different content and resolution are encoded by original HEVC algorithm. In order to balance the data quantity from different sequences, the training set is built by selecting every eight frames from the designated range in Table 1. According to the optimal depth or mode, each feature extracted from the training set are attached with tag 1 or tag 0 where the former represents normal HEVC coding while the latter represents fast coding of early termination (ET) or early split (ES). Taking the ETCU_S algorithm for example, if the optimal depth of CU is 2, all features of CU extracted at depth 0 and depth 1 are labeled 1. Namely CU at depth 0 and depth 1 should be split further. But for CU at depth 2, the extracted features are labeled 0 because it reaches the optimal depth. Afterwards the feature set is constructed by equivalently selecting the samples with tag 0 and tag 1. Then it is used to generate various classification trees. An example

TABLE 1. Training set.

Resolution	Sequence	Frames
2560×1600	PeopleonStreet	2-41
2560×1600	Traffic	2-41
1920×1080	Kimono1	2-81
1920×1080	ParkScene	2-81
832×480	PartyScene	2-201
416×240	BQSquare	2-321
1280×720	Johnny	2-121

of classification tree for the ETCU_S algorithm is illustrated in Fig. 5.

B. INTRA FEATURES

The intra features describe the spatial correlation among various blocks and modes in the inter coding.

1) The deviation of CU is represented by *DEV*,

$$DEV = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |P(i,j) - P\_avg| \quad (2)$$

where $P(i,j)$ is the luminance of each pixel and P_avg is the average of all $P(i,j)$ in the current CU. And W and H are the width and height of CU respectively. Small *DEV* indicates current CU is flat and smooth, the MSM mode is preferred in this situation.

2) The deviation of residual for the current CU is expressed by *DER*,

$$DER = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} |P_{res}(i,j) - P_{res}\_avg| \quad (3)$$

where $P_{res}(i,j)$ is the pixel residual in the current CU, and P_{res}_avg is the average of all $P_{res}(i,j)$ in the current CU. If *DER* is small, it indicates the prediction accuracy is high. Under this condition, current CU partition often adopts a large size because the prediction block is alike the original block. On the contrary, the CU tends to be divided into smaller size when the residual value is large. The distribution probability of feature *DER* for ET or non-ET of CU at depth 0 by encoding the training set with QP 32 is shown in Fig. 3. It can be observed that the distribution probability for CU with ET operation is much larger than those with non-ET operation when feature *DER* is less than 4.9. Otherwise CU with non-ET operation gains remarkable advantage over those with ET operation. Therefore, feature *DER* is suitable for the binary classification.

3) The difference between *DER* and *ADR* is expressed by *DDA*,

$$DDA = |DER - ADR| \quad (4)$$

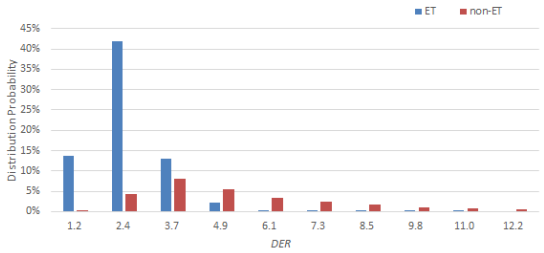

FIGURE 3. Distribution probability of the feature *DER*.

where *ADR* is defined by the average *DER* of four sub-CUs under the current CU,

$$ADR = \frac{1}{4} \sum_{k=0}^{3} (DER)_k, \quad k = 0, 1, 2, 3 \quad (5)$$

4) The sum of residual gradient is represented by *SRG*,

$$SRG = \sum_{i=1}^{W} \sum_{j=1}^{H} G_{res}(i,j) \quad (6)$$

where $G_{res}(i,j)$ is the gradient value at the coordinate (i,j) which is computed by performing the Sobel operator on the residual matrix, which is defined by $G_{res}(i,j)$,

$$G_{res}(i,j) = \left| \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} P_{res}(i,j) \right| + \left| \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} P_{res}(i,j) \right| \quad (7)$$

If the block contains some objects with complicated texture or inconsistent motion, *SRG* will become large. It hints that current CU is probably split into small CU. Otherwise the current CU will be coded with big size.

5) The distribution of TU coefficients shows that most of the non-zero coefficients concentrate on the above-left corner of TU. When the location of final non-zero coefficient approaches to the above-left corner, the current TU depth may be adopted as its optimal depth. Accordingly, the coordinates of final non-zero coefficients $final\_C_{NZ}(i,j)$ can indicate early termination for TU partition, which is denoted by *CNZ*,

$$CNZ = \{(i,j) | final\_C_{NZ}(i,j) \neq 0\} \quad (8)$$

6) If the non-zero coefficients of TU only spread in a limited scope, most of them will be the low-frequency coefficients. In this case, it is unnecessary to split TU because few coding bits are created by the following entropy coding. The number of non-zero coefficients is obtained by *NNC*,

$$NNC = \sum_{i} \sum_{j} C_{NZ}(i,j) \quad (9)$$

where $C_{NZ}(i,j)$ is equal to 1 when the corresponding coefficient is not zero, otherwise $C_{NZ}(i,j)$ is equal to 0.

7) The energy of TU can be calculated by *ENG*,

$$ENG = \sum_{i=1}^{W}\sum_{j=1}^{H} C^2(i,j) \tag{10}$$

where $C(i,j)$ is the quantized coefficient. If the residual blocks are evenly distributed, *ENG* is small because most of $C(i,j)$ is zero. Consequently the current TU need not be divided further in this situation.

8) The average depth of spatially adjacent CUs is denoted by *AVD*,

$$AVD = \frac{1}{3}\left(Dep_{abv} + Dep_{lef} + Dep_{abl}\right) \tag{11}$$

where Dep_{abv}, Dep_{lef}, Dep_{abl} are the depths of above CU, left CU and above-left CU associated with current CU respectively.

9) The average mode value for spatial neighboring PU is expressed by *AVM*,

$$AVM = \frac{1}{3}[PU_{abv}(i,j)+PU_{lef}(i,j)+PU_{abl}(i,j)] \tag{12}$$

where PU_{abv}, PU_{lef}, PU_{abl} are the optimal PU mode of above CU, left CU and above-left CU associated with current CU respectively.

10) The Boolean operator denotes whether the optimal mode is SKIP after all 2N×2N modes are coded. It is expressed by *MSP*,

$$MSP = Bool(\text{Mode}_{2N\times 2N} == \text{SKIP}) \tag{13}$$

For SKIP mode, little information is encoded because only the mode tag is transmitted. If SKIP mode is selected as the optimal PU mode after all 2N×2N modes are coded, early termination for PU mode decision may be feasible. For the CU with simple texture, slow motion or stationary objects, SKIP mode is often selected as the optimal PU mode.

11) The ratio of *RDC* is represented by *RRC*,

$$RRC = \frac{RDC_{MSM}}{RDC_{2N\times 2N}} \tag{14}$$

where RDC_{MSM} is the RD cost of MSM mode, and $RDC_{2N\times 2N}$ is the RD cost of 2N×2N mode. This feature is also used in [20]. Because only mode tags, index or residual data are encoded in MSM mode, its RD cost is less than other modes. When *RRC* is small, it specifies current CU is inclined to stop partition in advance.

12) The ratio of *RDC* and distortion for MSM mode is denoted by *RRD*,

$$RRD = \frac{RDC_{MSM}}{D_{MSM}} \tag{15}$$

where RDC_{MSM} and D_{MSM} are the RD cost and distortion for MSM mode respectively. The feature *RRD* is small when MSM is selected as the best PU mode. In addition, Fig. 4 illustrates the distribution probability of feature *RRD* for ET or non-ET of CU at depth 0 by encoding the training set with QP 32. It can be observed that the ET operation will be applied to most of CU if *RRD* is less than 1.4. Otherwise, the non-ET operation is preferred for the remaining CU. Therefore, the feature *RRD* is also fit for binary classification.

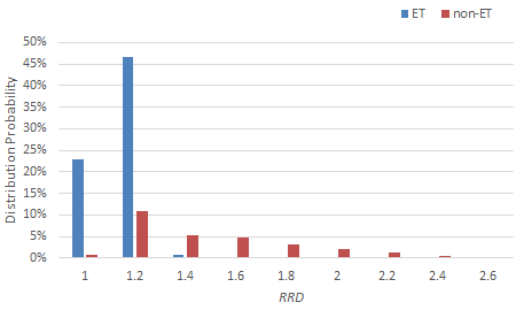

FIGURE 4. Distribution probability of the feature ***RRD***.

C. INTER FEATURES

Temporal correlation specified by inter features is an important clue for fast coding. If current CU is verified similar to the co-located CU, current CU determination may be early stopped.

1) The normalization of difference of distortion between temporal adjacent CU is expressed by *NDD*,

$$NDD = \left|\log_{10}\frac{|D_{cur} - D_{col}|}{D_{col}}\right| \tag{16}$$

where D_{cur} and D_{col} is the distortion of current CU and its co-located CU in the reference frame respectively.

2) The difference of feature *RRD* between temporal adjacent CU is measured by *DRD*,

$$DRD = |RRD_{cur} - RRD_{col}| \tag{17}$$

3) The difference of sum of gradient elements between temporally adjacent CU is measured by *DSG*,

$$DSG = |G_{cur}\_sum - G_{col}\_sum| \tag{18}$$

where G_{cur}_sum and G_{col}_sum are the sum of gradient of current CU and co-located CU respectively. They are both calculated by,

$$G\_sum = \sum_{i=1}^{W}\sum_{j=1}^{H} G(i,j) \tag{19}$$

where $G(i,j)$ is the gradient matrix computed by the Sobel operator, which is similar to (7).

4) The normalization of difference of deviation between the current CU and its co-located CU is calculated by *NDV*,

$$NDV = \frac{\log_{10}|DEV_{cur} - DEV_{col}|}{\log_{10} DEV_{col}} \tag{20}$$

D. CLASSIFICAITON TREES

Both intra features and inter features are applied to generate the classification trees by the CART algorithm. The main merit of CART algorithm is that it can automatically choose the continuous and discrete features and build the interactions among them. When it is used to generate the classification tree, the essential features will be identified and the insignificant features will be neglected. These merits make it efficient to select proper features to build classification tree. In this paper, the Gini index [28] is used to select the attribute of node partition, which is defined by,

$$G = 1 - \sum_{i=1}^{m} p_i^2 \tag{21}$$

where p_i is the percentage that sample quantity of category i accounts for total sample quantity. Gini index denotes the impurity that one category is wrongly assigned in the decision procedure. When Gini index becomes smaller, the classification tree becomes more accurate. When constructing a classification tree, the Gini index for all available splits at each internal node are computed by the CART algorithm, then the feature and its threshold corresponding to the split with the minimum Gini index is adopted in the classification tree. However, the resulting classification tree usually has high complexity because of the over-fitting problem. Therefore, the original classification trees have to be pruned before it is applied to the fast coding.

Because early termination and early split are used in the proposed algorithm, the binary classification at each node is defined that the tag 1 represents the normal HEVC coding while the tag 0 represents early termination or early split in different algorithms. Because large amount of feature samples can result in high complicated classification tree, additional constraints are necessary in the proposed algorithm. The maximum depth of classification trees is set to 10. The sample quantity for each parent node and each child node is restricted by no less than 1% and 0.5% of the quantity of total samples respectively. In addition, the credible rate that the samples from parent node are correctly allocated to the child node with tag 0 is calculated by collating their initial properties. And the credible threshold is utilized to prune the classification tree. The child node will become the leaf node if its credible rate is larger than the credible threshold. Otherwise the child node is still an intermediate node and continues to be divided further. Higher credible threshold leads to more accurate classification tree but decreases the time gains in the fast coding algorithm. Considering the difference among the individual sub-algorithm, the credible threshold is defined 85% for the ETCU_T algorithm, 70% for the ETTU algorithm and 80% for the remaining algorithms. An example of classification tree for ETCU_S algorithm at depth 2 with QP 32 under random access (RA) configuration is shown in Fig. 5. It illustrates that the classification tree is built by recursively deciding the node partition according to different features. At each internal node, the feature and its

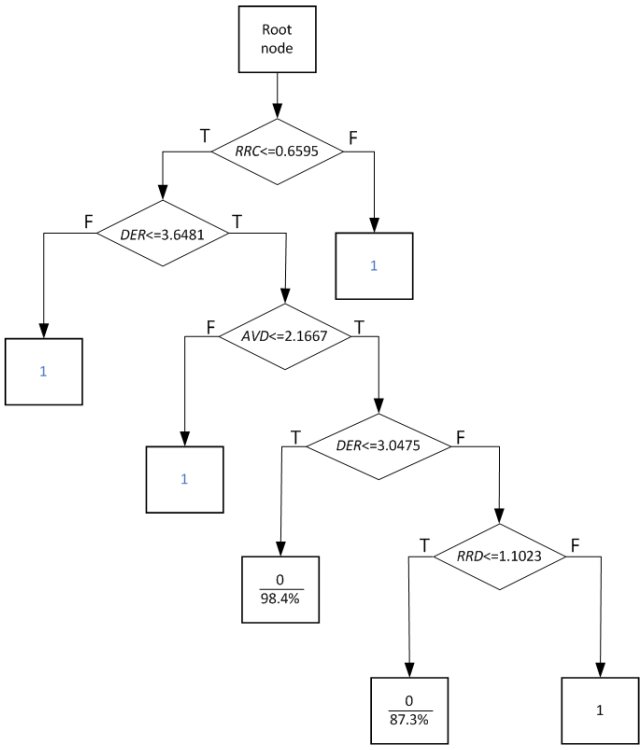

FIGURE 5. An example of classification tree.

threshold which results in the minimal Gini index is determined by the CART algorithm. Beginning with root node, it is divided into two child nodes by the first judgement $RRC <= 0.6595$. When it is false (F), the child node is assigned with tag 1 which denotes that current CU will be split according to the normal HEVC algorithm. Conversely if it is true (T), it continues to the second judgement $DER <= 3.6418$. Similarly current CU will be split by the normal HEVC algorithm when the judgement is false. Otherwise it turns to next judgement $AVD <= 2.1667$. Until when the fourth judgement $DER <= 3.0475$ is true, the child node with tag 0 becomes a leaf node and its following nodes are pruned because its credible rate 98.4% is larger than the credible threshold. In this situation, current CU partition is early terminated and the optimal depth is decided. The classification process continues until the fifth judgement $RRD <= 1.1023$ is completed. At the bottom of classification tree, there is another leaf node which satisfies the condition of early termination. It can be seen in the Fig. 5, classification tree can build up clear interconnection among different features, which is easy to be implemented.

E. FAST CODING ALGORITHM

The fast coding algorithm at CU level consists of three sub-algorithms ETCU_T, ESCU and ETCU_S. At first, the ETCU_T algorithm is performed at CU depth 0 or 1. Temporal correlation between the current frame and its reference frame for sequence BasketballPass is shown in Fig. 6, where most of background region characterized by simple texture

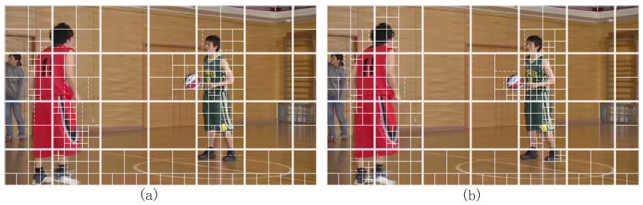

FIGURE 6. An example of temporal similarity in the inter coding.

TABLE 2. The similarity between current CU and its co-located CU.

CU Depth	0	1	2	3
Similarity	91.9%	84.2%	60.2%	10.5%

and slow variation is encoded by large CU in both frames. The similarity that the current CU and its co-located CU in sequence BasketballPass have same depth under LDP configuration is given in Table 2. It can be seen that the current CU and co-located CU have high similarity at depth 0 and 1. Consequently, if current block belongs to the background meanwhile its co-located block is coded as depth 0 or 1 in the reference frame, it is highly probable that current block will adopt the same depth as the co-located CU. Under this condition, early termination of CU partition can be applied at depth 0 or 1. Here the feature *NDD, DRD, DSG* and *NDV* are used for ETCU_T algorithm. Afterwards the ESCU algorithm is used. As the region with complicated texture or fast motion is often coded by small size block, the coding time can be saved if CU decision at the small depth is skipped. The features *DER, DDA* and *SRG* are used for ESCU method. Finally different from the ETCU_T algorithm depending on the inter features, the ETCU_S algorithm based on the intra features *AVD, DER, RRC* and *RRD* is performed to decide the early termination of CU partition by referring the spatially adjacent CU.

At each CU level, the ETPU algorithm is executed. As shown in Fig. 2, the MSM mode accounts for the highest distribution probability. Hence the MSM mode and non-MSM mode are assorted at first. If MSM mode is selected as the best mode of PU, the remaining decision for SMP mode and AMP mode is skipped. Otherwise, the non-MSM mode will continued to be divided into SMP mode and AMP mode. If the SMP mode is chosen, the decision for AMP mode will be skipped. The static or slow changing block is often coded by large CU, where the inter-predication residuals are small and the coding coefficients are also sparse after transformation and quantification. Therefore, the RD costs are small for these blocks. In this situation, MSM mode is often adopted to save coding bits. On the other side, for fast moving block, the SMP or AMP mode may be applied in order to accurately describe the complicated motion mode. Consequently, the feature *AVD, AVM, DEV, MSP* and *RRD* are used to decide the PU mode in the ETPU algorithm.

Associated to each CU level, the fast coding algorithm ETTU is fulfilled. After inter prediction, transformation coding is performed on the residual blocks which are divided into different size based on RQT structure. Additional complexity is introduced by the iterative decision process of RQT as well as the integer transformation and scanning mode for the transformation coefficients. If the residual blocks are evenly distributed in spatial domain, most of the quantized transformation coefficients are 0. At this case, TU is subject to big size. On the contrary, if the residual block has rich texture, TU is probably divided further because the quantization coefficients become more complicated. Because of the fact that entropy coding includes the position coding and the amplitude coding of non-zero coefficient, the position of the final non-zero coefficient in TU is very important because the trailing zero coefficients can be omitted. In addition, the non-zero coefficients often spread in the narrow range if it is flat or gradually varied inside the TU. According to these principles, the feature *CNZ, NNC* and *ENG* are used for the ETTU algorithm.

F. THE FRAMEWORK OF PROPOSED ALGORITHM

At last, the framework of proposed algorithm is illustrated in Fig. 7. Firstly the classification features including intra features and inter features are extracted by encoding the training set. Then the classification trees are created by using these features. Based on them, five sub-algorithms are performed on CU, PU and TU respectively. After starting to encode a CTU, the ETCU_T algorithm is performed. If the judgement of ET is No, the ESCU algorithm is carried out. Otherwise the CU split is early stopped and shifted to the optimal depth decision. If the judgement of ES is Yes, the following calculation of RD cost for mode selection is skipped and current CU can be directly split. Otherwise, the program advances to the ETPU algorithm. Meanwhile the ETTU algorithm is executed. Afterwards the ETCU_S algorithm is used to decide whether CU will be split further. At the end of the whole algorithm, the optimal depth is decided for CU.

IV. EXPERIMENTAL RESULTS

In our experiments, the recent version of HM16.9 and two popular configurations including LDP and RA are used. In the LDP configuration, the coding frames consist of the first I frame followed by P prediction frames. Since the sequence of coding frame maintains invariant, the coding delay is low. While in the RA configuration, the coding efficiency is enhanced by the bi-directional hierarchical prediction structure. But it undergoes large coding delay because of the frame reordering. All simulation setting is conformed to the common HM test conditions and software reference [29]. The Windows 7 operating system and Intel i7-6700 processor at 3.4GHz is used for the simulation platform.

Total 18 sequences belonging to 5 classes are used to evaluate the proposed algorithm, which is listed in Table 3. And the coding complexity is measured by the average encoding time saving (TS) under four QPs including 22, 27, 32 and 37,

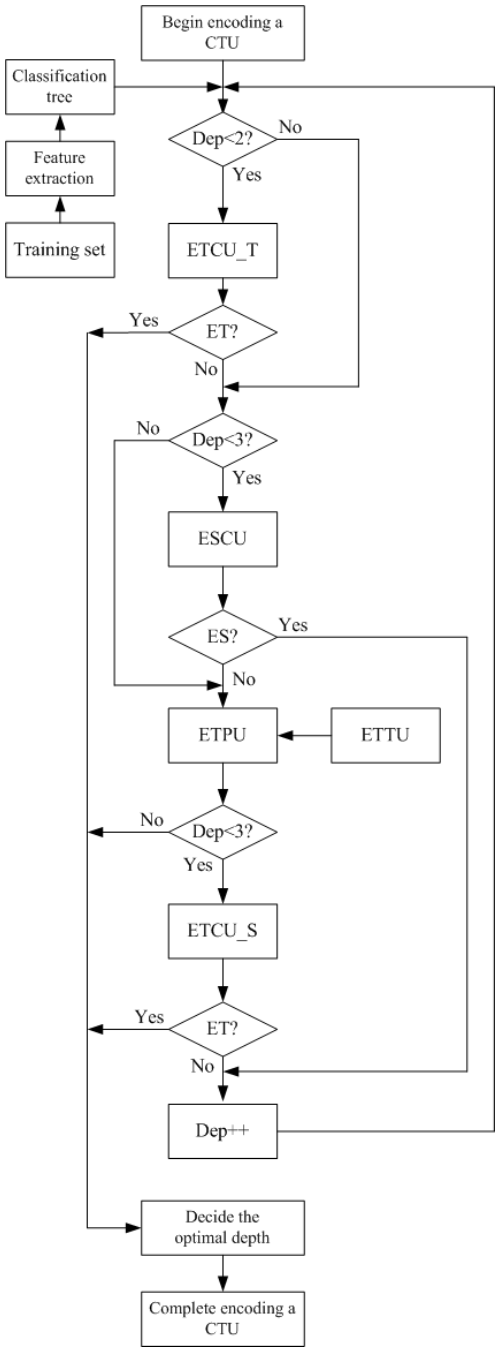

FIGURE 7. The framework of proposed algorithm.

which is formulated as,

$$TS = \frac{1}{4}\sum_{i=1}^{4}\frac{T_{org\_i} - T_{prop\_i}}{T_{org\_i}} \times 100\% \tag{22}$$

where T_{prop_i} denotes the encoding time by using the proposed algorithm and T_{org_i} represents the encoding time by using the original HM16.9 algorithm. The average coding efficiency is measured by the BDBR and BDPSNR [30]. Because of the limited space, BDBR and BDPSNR are abbreviated as BR and BP respectively. The average performance for each class and the total average performance are abbreviated as Avg and T_Avg respectively.

TABLE 3. Test sequences.

Number	Sequence	Class	Resolution	Frames
S01	PeopleOnStreet	A	2560×1600	150
S02	Traffic	A	2560×1600	150
S03	BasketballDrive	B	1920×1080	500
S04	BQTerrace	B	1920×1080	600
S05	Cactus	B	1920×1080	500
S06	Kimino1	B	1920×1080	240
S07	ParkScene	B	1920×1080	240
S08	BQMall	C	832×480	600
S09	PartyScene	C	832×480	500
S10	RaceHorsesC	C	832×480	300
S11	BasketballDrill	C	832×480	500
S12	BasketballPass	D	416×420	500
S13	BlowingBubbles	D	416×420	500
S14	BQSquare	D	416×420	600
S15	RaceHorses	D	416×420	300
S16	FourPeople	E	1280×720	600
S17	Johnny	E	1280×720	600
S18	KristenAndSara	E	1280×720	600

TABLE 4. Coding performance at three-level [unit:%/%].

SEQUENCES	CU LEVEL		PU LEVEL		TU LEVEL	
	BR	TS	BR	TS	BR	TS
PeopleOnStreet(A)	1.31	34.6	0.05	12.1	0.11	4.2
Traffic(A)	0.74	54.1	0.09	30.9	0.09	7.4
BQTerrace(B)	0.19	45.9	0.13	30.0	0.26	9.7
Kimino1(B)	0.88	48.9	0.08	27.6	0.15	6.9
ParkScene(B)	0.19	49.1	0.09	28.3	0.24	7.1
BQMall(C)	0.98	44.9	0.10	19.9	0.21	7.0
PartyScene(C)	0.61	32.6	0.03	10.6	0.25	7.6
BasketballDrill(C)	0.85	39.0	0.02	16.0	0.03	5.8
BasketballPass(D)	0.88	35.7	0.07	13.3	0.15	5.5
BlowingBubbles(D)	0.50	29.3	0.01	12.6	0.27	8.6
BQSquare(D)	0.19	31.6	0.04	19.0	0.32	9.2
FourPeople(E)	0.50	65.1	-0.01	40.4	-0.01	9.9
Johnny(E)	0.16	71.3	0.04	46.7	-0.01	10.9
KristenAndSara(E)	0.48	68.0	0.10	43.3	0.06	9.3
T_Avg	0.60	46.4	0.06	25.1	0.15	7.8

A. OBJECTIVE EVALUATION

At first, the coding performance at three levels including CU, PU and TU under RA configuration is given in Table 4. It can be found that the fast algorithm at CU level reaches the highest time saving of 46.4%. It is because early termination or split of CU will skip all associated mode decision, which can be seen in Fig. 7. And the fast algorithm at TU level endures the limited time saving because the original coding time for RQT module accounts for small portion in the whole coding. The BDBR is comparatively higher at TU level because lower credible threshold is used in the classification

TABLE 5. Performance comparison between popular algorithms and proposed algorithm under LDP configuration [unit:%/dB/%].

SEQ.	MSVM[18]			AOM[9]			AVO[10]			BML[23]			FSVM[24]			Proposed		
	BR	BP	TS	BR	BP	TS	BR	BP	TS	BR	BP	TS	BR	BP	TS	BR	BP	TS
S01(A)	1.40	-0.05	43.4	0.28	-0.04	24.8	5.14	-0.17	68.1	3.63	-0.12	60.8	—	—	—	1.49	-0.07	37.3
S02(A)	1.91	-0.05	56.7	0.39	-0.02	35.9	5.41	-0.13	73.4	5.67	-0.13	74.7	—	—	—	1.63	-0.05	55.2
Avg	1.66	-0.05	50.1	0.33	-0.03	30.4	5.28	-0.15	70.8	4.65	-0.12	67.8	—	—	—	1.56	-0.06	46.3
S03(B)	2.59	-0.04	46.3	1.10	-0.03	33.1	5.86	-0.10	43.5	5.98	-0.09	74.0	3.45	-0.06	59.4	1.80	-0.05	49.2
S04(B)	1.17	-0.02	45.6	0.90	-0.04	33.0	3.34	-0.05	46.5	5.82	-0.05	67.6	2.20	-0.04	58.7	0.89	-0.02	49.9
S05(B)	1.62	-0.02	46.9	1.52	-0.04	30.5	4.85	-0.07	46.6	4.11	-0.07	68.3	3.22	-0.05	59.3	1.71	-0.04	48.0
S06(B)	2.14	-0.06	48.1	1.31	-0.05	34.0	4.18	-0.12	40.6	2.96	-0.09	71.1	2.58	-0.07	60.5	1.71	-0.06	49.0
S07(B)	1.42	-0.03	45.5	0.55	-0.02	29.0	3.71	-0.10	46.4	4.16	-0.09	67.8	2.71	-0.07	56.5	1.65	-0.05	48.7
Avg	1.79	-0.03	46.5	1.08	-0.03	31.9	4.39	-0.09	44.7	4.61	-0.08	69.8	2.83	-0.06	58.9	1.55	-0.04	49.0
S08(C)	1.79	-0.06	40.4	0.74	-0.03	17.2	4.90	-0.16	48.1	—	—	—	3.74	-0.12	55.7	1.60	-0.06	43.9
S09(C)	1.00	-0.02	33.3	2.34	-0.12	20.3	2.39	-0.08	53.3	2.40	-0.08	55.7	2.20	-0.08	52.5	1.27	-0.05	33.6
S10(C)	1.52	-0.05	37.1	1.41	-0.07	23.5	4.27	-0.13	52.7	2.51	-0.07	54.4	2.50	-0.07	51.1	1.16	-0.04	35.1
S11(C)	1.55	-0.05	45.4	0.66	-0.03	28.0	5.17	-0.18	44.2	3.79	-0.11	63.7	3.35	-0.11	58.9	1.22	-0.05	42.5
Avg	1.46	-0.04	39.1	1.29	-0.06	22.3	4.18	-0.14	49.6	2.90	-0.09	57.9	2.95	-0.09	54.6	1.31	-0.05	38.8
S12(D)	1.11	-0.05	36.8	0.61	-0.04	21.3	3.96	-0.17	43.3	—	—	—	2.93	-0.11	55.9	1.13	-0.05	36.5
S13(D)	0.74	-0.02	28.8	1.35	-0.07	19.4	2.94	-0.09	46.3	2.08	-0.06	51.0	2.49	-0.08	51.4	1.25	-0.05	32.7
S14(D)	0.93	-0.03	29.6	1.30	-0.06	27.0	2.81	-0.09	49.9	3.83	-0.09	61.6	1.69	-0.05	53.3	1.08	-0.04	34.9
S15(D)	0.86	-0.03	28.4	2.16	-0.12	29.2	3.83	-0.14	52.4	2.81	-0.11	52.0	2.71	-0.10	47.9	1.00	-0.04	30.2
Avg	0.91	-0.03	30.9	1.35	-0.07	24.2	3.39	-0.12	48.0	2.91	-0.09	54.9	2.46	-0.09	52.1	1.12	-0.05	33.6
S16(E)	1.87	-0.05	67.3	-0.35	0.00	44.6	3.42	-0.09	38.2	4.91	-0.11	83.0	2.81	-0.07	70.3	2.20	-0.08	69.8
S17(E)	2.49	-0.04	68.6	-0.49	0.00	43.0	4.24	-0.07	36.4	5.28	-0.09	82.6	3.91	-0.06	69.8	2.25	-0.05	77.1
S18(E)	1.86	-0.04	67.9	-0.25	0.00	42.5	3.83	-0.09	35.8	4.79	-0.11	82.0	2.96	-0.07	69.6	1.54	-0.05	73.3
Avg	2.07	-0.05	67.9	-0.37	0.00	43.4	3.83	-0.08	36.8	4.99	-0.10	82.5	3.23	-0.07	69.9	2.00	-0.06	73.4
T_Avg	1.55	-0.04	45.3	0.86	-0.04	29.8	4.12	-0.11	48.1	4.05	-0.09	66.9	2.84	-0.08	58.2	1.48	-0.05	47.1

TABLE 6. Performance comparison between popular algorithms and proposed algorithm under RA configuration [unit:%/dB/%].

SEQ.	MSVM[18]			AOM[9]			AVO[10]			BML[23]			FSVM[24]			Proposed		
	BR	BP	TS	BR	BP	TS	BR	BP	TS	BR	BP	TS	BR	BP	TS	BR	BP	TS
S01(A)	1.76	-0.06	47.4	-0.05	-0.02	20.7	4.45	-0.16	54.8	3.02	-0.11	57.9	2.84	-0.11	47.2	1.55	-0.07	40.2
S02(A)	1.94	-0.05	61.6	0.05	-0.01	33.0	3.13	-0.10	38.3	4.49	-0.11	71.3	2.00	-0.06	60.3	1.11	-0.04	62.3
Avg	1.85	-0.06	54.5	0.00	-0.01	26.9	3.79	-0.13	46.6	3.76	-0.11	64.6	2.42	-0.08	53.8	1.33	-0.06	51.3
S03(B)	4.84	-0.08	51.7	0.65	-0.02	27.7	4.45	-0.07	38.6	5.23	-0.08	69.9	3.81	-0.06	57.0	1.64	-0.04	55.0
S04(B)	1.38	-0.02	49.9	0.36	-0.01	33.0	2.15	-0.04	38.7	2.25	-0.03	63.3	1.54	-0.03	59.8	1.04	-0.02	58.1
S05(B)	2.29	-0.04	51.2	0.43	-0.01	24.5	3.69	-0.06	40.6	4.25	-0.06	66.0	3.00	-0.05	56.7	1.11	-0.02	53.8
S06(B)	4.93	-0.13	52.5	1.25	-0.04	27.2	2.88	-0.09	34.8	3.66	-0.09	65.3	2.36	-0.07	57.1	1.32	-0.04	56.4
S07(B)	1.72	-0.05	53.2	0.29	-0.01	26.1	2.81	-0.08	39.9	3.16	-0.08	67.2	2.26	-0.06	58.3	1.06	-0.03	58.5
Avg	3.03	-0.06	51.7	0.60	-0.02	27.7	3.20	-0.07	38.5	3.71	-0.07	66.3	2.59	-0.05	57.8	1.23	-0.03	56.4
S08(C)	2.64	-0.09	47.7	-0.03	-0.01	16.3	4.56	-0.17	43.5	—	—	—	3.85	-0.13	55.8	1.43	-0.06	52.5
S09(C)	1.10	-0.04	40.7	0.91	-0.05	18.7	1.81	-0.08	47.0	1.56	-0.07	55.6	1.93	-0.08	54.0	0.91	-0.04	40.8
S10(C)	3.21	-0.11	41.3	1.01	-0.05	20.6	4.37	-0.14	50.6	3.13	-0.11	49.4	3.03	-0.09	51.4	1.35	-0.05	38.0
S11(C)	1.92	-0.06	50.7	0.08	-0.01	21.8	3.73	-0.15	39.6	2.29	-0.07	59.1	3.12	-0.11	57.6	0.97	-0.04	46.0
Avg	2.22	-0.07	45.1	0.49	-0.03	19.4	3.62	-0.13	45.2	2.33	-0.08	54.7	2.98	-0.10	54.7	1.17	-0.05	44.3
S12(D)	2.73	-0.13	40.5	0.10	-0.01	21.4	3.03	-0.14	40.2	—	—	—	3.28	-0.14	55.8	1.14	-0.05	41.6
S13(D)	0.99	-0.03	33.7	0.61	-0.03	15.7	2.49	-0.09	41.6	1.37	-0.05	50.2	2.51	-0.09	52.4	0.78	-0.03	39.5
S14(D)	0.64	-0.03	35.3	0.06	-0.01	23.3	1.25	-0.05	40.0	0.94	-0.05	52.3	1.26	-0.06	58.3	0.54	-0.02	44.8
S15(D)	2.18	-0.10	33.1	1.43	-0.08	25.0	3.81	-0.16	50.9	1.89	-0.08	47.2	3.29	-0.14	48.3	0.95	-0.04	32.9
Avg	1.64	-0.07	35.7	0.55	-0.03	21.4	2.64	-0.11	43.2	1.40	-0.06	49.9	2.59	-0.11	53.7	0.85	-0.04	39.7
S16(E)	1.24	-0.04	71.3	-0.13	0.00	42.5	2.44	-0.08	83.6	3.72	-0.11	83.8	—	—	—	0.65	-0.03	74.5
S17(E)	1.53	-0.03	72.3	-0.18	0.00	42.0	3.57	-0.07	84.0	7.45	-0.14	83.2	—	—	—	0.36	-0.01	80.9
S18(E)	1.33	-0.04	71.6	-0.21	0.01	43.2	2.99	-0.09	83.2	6.35	-0.17	84.1	—	—	—	0.79	-0.03	77.4
Avg	1.37	-0.04	71.7	-0.17	0.00	42.6	3.00	-0.08	83.6	5.84	-0.14	83.7	—	—	—	0.60	-0.02	77.6
T_Avg	2.13	-0.06	50.3	0.37	-0.02	26.8	3.20	-0.10	49.4	3.42	-0.09	64.1	2.67	-0.09	55.3	1.04	-0.04	53.0

tree for the ETTU algorithm in order to attain the reasonable time saving.

The coding performance for the whole fast algorithm in comparison with several popular algorithms are enumerated in Table 5 and 6 under the configuration of LDP and RA respectively. In both tables, the average performance for each class of sequence and all sequences is calculated respectively. The comparative algorithms contain the multiple SVM classifier (MSVM) based algorithm [18], adaptive ordering of modes (AOM) based algorithm [9], adaptive visiting order (AVO) based algorithm [10], binary and multi-class learning (BML) based algorithm [23] and fuzzy SVM (FSVM) based

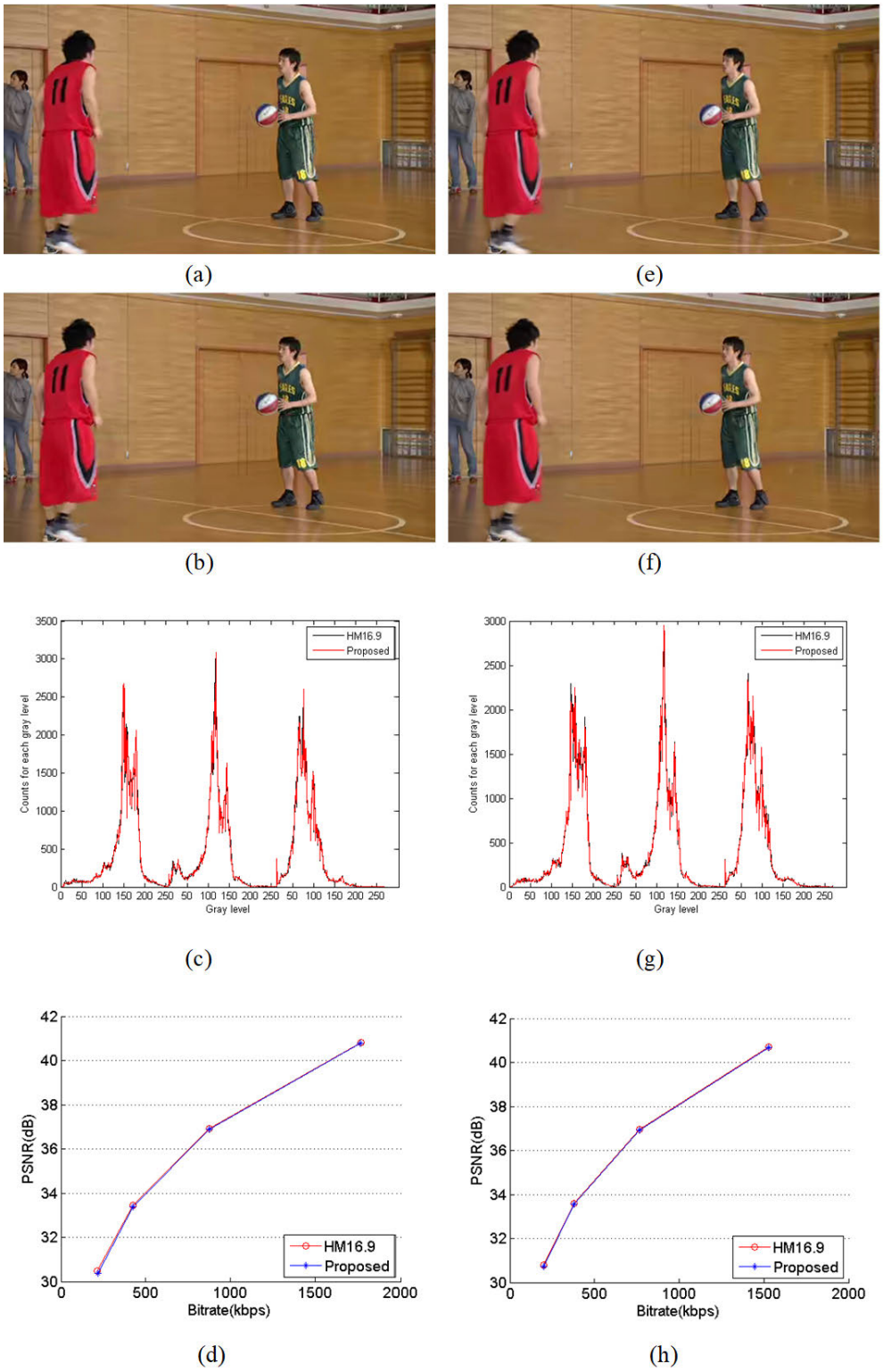

FIGURE 8. Subjective evaluation for sequence BasketballPass. (a)the decoded frame by HM16.9 under LDP configuration, (b)the decoded frame by the proposed algorithm under LDP configuration, (c)the histogram for figure (a) and (b), HS = 97.7%, (d)the RD curve for the HM16.9 and the proposed algorithm under LDP configuration; (e)the decoded frame by HM16.9 under RA configuration, (f)the decoded frame by the proposed algorithm under RA configuration, (g)the histogram for figure (e) and (f), HS = 98%, (h)the RD curve for the HM16.9 and the proposed algorithm under RA configuration.

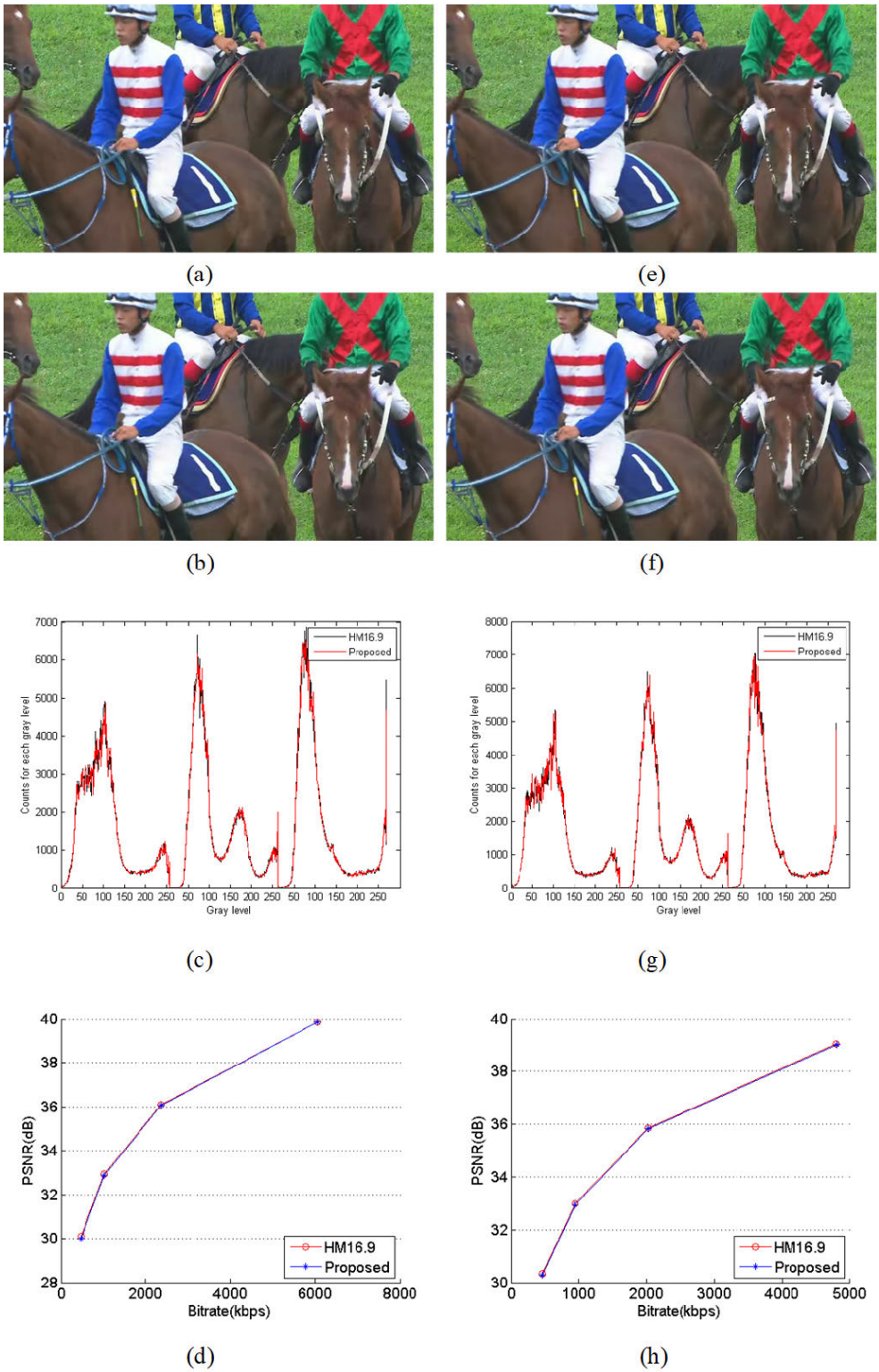

FIGURE 9. Subjective evaluation for sequence RacehorsesC. (a)the decoded frame by HM16.9 under LDP configuration, (b)the decoded frame by the proposed algorithm under LDP configuration, (c)the histogram for figure (a) and (b), HS = 97.9%, (d)the RD curve for the HM16.9 and the proposed algorithm under LDP configuration; (e)the decoded frame by HM16.9 under RA configuration, (f)the decoded frame by the proposed algorithm under RA configuration, (g)the histogram for figure (e) and (f), HS = 98.1%, (h)the RD curve for the HM16.9 and the proposed algorithm under RA configuration.

algorithm [24]. In terms of average performance for each class in Table 5, the class E achieves the largest time saving for most of the algorithms. Because of simple content and slow motion in the sequences of class E, small depths are adopted by most of CU partition. It leads to easy accomplishment of early termination algorithm, which brings about large amount of time saving. On the contrary, most of the fast algorithms are inefficient for class D. One reason is that these sequences are composed of complication texture and fast motion. The other reason is that original coding time for these sequences is less than other sequences because their resolutions are the smallest in all sequences. Observing the total average performance in Table 5, the BML [23] algorithm achieves the maximum time saving of 66.9% while the minimum time saving of 29.8% occurs in the AOM [9] algorithm. The second highest time saving of 58.2% is attained by the FSVM [24] algorithm but it attains 1.21% lower BDBR than the BML [23] algorithm. In addition, the AOM [9] algorithm obtains the lowest BDBR of 0.86% while the BML [23] algorithm endures the high BDBR of 4.05%. The proposed algorithm have the similar time efficiency to the MSVM [18] algorithm and the AVO [10] algorithm, but it achieves 0.07% and 2.64% lower BDBR than the two algorithms respectively. It is also found that the BDPSNR for the MSVM [18] algorithm, the AOM[9] algorithm and the proposed algorithm are approximately -0.05dB, which is higher than the other three algorithms. According to the average performance for each class in Table 6, the maximum time saving still appears in class E and the minimum time saving also occurs in class D for most of fast algorithms. Watching the total average performance in Table 6, both the maximum time saving of 64.1% and the highest BDBR of 3.42% are attained by the BML [23] algorithm. Although the AOM [9] algorithm suffers the minimum time saving of 26.8% but achieves the lowest BDBR of 0.37% and attains the highest BDPSNR of -0.02dB in all algorithms. Resembling the situation in Table 5, the second highest time saving of 55.3% is attained by the FSVM [24] algorithm but it attained 0.75% lower BDBR than the BML [23] algorithm. It is also found that the proposed algorithm is superior to the MSVM [18] algorithm and the AVO [10] algorithm in time efficiencies. Namely it achieves 2.7% and 3.6% higher time saving than the two algorithms respectively. Meanwhile, the BDBR of proposed algorithm decreases 1.09% and 2.16% in comparison with the two algorithms respectively. Furthermore, the proposed algorithm gains 0.02dB and 0.06dB higher BDPSNR than the two algorithms respectively. Another merit for the proposed algorithm in Table 6 is that the BDBR for each sequence is lower than 2%. It indicates that the proposed algorithm can attain the favorable time efficiency meanwhile control the compression loss well. The experimental results also show that the promotion of time saving is accompanied by the degradation of BDBR and BDPSNR. Consequently the trade-off between time efficiency and coding quality usually depends on different algorithms in different applications. Considering that the video quality becomes more significant for many viewers in recent years, the proposed algorithm is designed to obtain good overall performance with little compression loss.

B. SUBJECTIVE EVALUATION

The average BDPSNR for the proposed algorithm under the configuration of LDP and RA is −0.05dB and −0.04dB respectively, which are small enough not to injure the video quality. The decline of BDPSNR is due to the fact that some non-optimal modes in the original HEVC algorithm are judged as the optimal modes in the proposed algorithm. On the other hand, BDPSNR is often considered as an objective evaluation method, which is uncertain to be consistent to the subjective evaluation. In order to demonstrate the subjective quality of proposed algorithm, the histogram similarity is used to measure the disparity between the HM16.9 and the proposed algorithm. In Fig. 8 and Fig. 9, both the 10th frame is selected to verify the subjective quality from two representative sequences of BasketballPass and RacehorsesC. Both the decoded frames under LDP and RA configuration are illustrated for each sequence. In Fig. 8(a) and Fig. 8(b), no matter in the region with rich texture or flat texture, the subjective perception of the decoded frame is almost the same between the proposed algorithm and the original HM16.9 algorithm. Similar conclusion can be drawn for Fig. 8(e) and Fig. 8(f), Fig. 9(a) and Fig. 9(b) as well as Fig. 9(e) and Fig. 9(f). Furthermore, the histograms for Fig. 8(a) and Fig. 8(b) are nearly overlapped in Fig. 8(c) where three waveforms represent three histograms for color channel R, G and B respectively. Each one ranges from 0 to 255 because all testing sequences have 8-bit color depth. Similar definitions and results can be seen in Fig. 8(g), Fig. 9(c) and Fig. 9(g). We use histogram similarity (HS) to measure these results, which is defined as following,

$$HS = 1 - \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=0}^{n} \frac{|H_P(i,j) - H_{HM}(i,j)|}{\text{Res\_seq}} \quad (23)$$

where m = 3 and n = 255. $H_{HM}(i,j)$ is the histogram for HM16.9 while $H_P(i,j)$ is the histogram for the proposed algorithm. Res_seq is the resolution of sequence. HS reaches 97.7% and 98% for sequence BasketballPass as well as 97.9% and 98.1% for sequence RacehorsesC under LDP and RA configuration respectively. The final rows in both Fig. 8 and Fig. 9 exhibit that the RD curves for HM16.9 and the proposed algorithm match well, which indicates the proposed algorithm can maintain compression efficiency and video quality well.

V. CONCLUSION

In this paper, the intra features and inter features related to fast decision for CU, PU and TU are exploited. Based on these features, the classification trees are constructed by carefully selecting features and designing the classification criteria. By analyzing the spatial and temporal correlation, a fast coding algorithm based on hierarchical classification

tree is proposed. Experimental results show that the proposed algorithm can achieve good overall performance by balancing time efficiency and compression efficiency. Particularly it is demonstrated that the proposed algorithm causes little quality loss in terms of objective evaluation by BDPSNR and subjective evaluation by the histogram similarity.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] R. Duvar, O. Akbulut, and O. Urhan, "Fast inter mode decision exploiting intra-block similarity in HEVC," *Signal Process., Image Commun.*, vol. 78, pp. 503–510, Oct. 2019.

[3] J. Lei, J. Duan, F. Wu, N. Ling, and C. Hou, "Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3D-HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 706–718, Mar. 2018.

[4] J. H. Bae and M. H. Sunwoo, "Adaptive early termination algorithm using coding unit depth history in HEVC," *J. Signal Process. Syst.*, vol. 91, no. 8, pp. 863–873, Jul. 2018.

[5] T.-H. Tsai, S.-S. Su, and T.-Y. Lee, "Fast mode decision method based on edge feature for HEVC inter-prediction," *IET Image Process.*, vol. 12, no. 5, pp. 644–651, May 2018.

[6] D. G. Fernández, A. A. Del Barrio, G. Botella, and C. García, "Fast and effective CU size decision based on spatial and temporal homogeneity detection," *Multimedia Tools Appl.*, vol. 77, no. 5, pp. 5907–5927, Feb. 2017.

[7] K. Duan, P. Liu, K. Jia, and Z. Feng, "An adaptive quad-tree depth range prediction mechanism for HEVC," *IEEE Access*, vol. 6, pp. 54195–54206, 2018.

[8] J. Tariq, S. Kwong, and H. Yuan, "Spatial/temporal motion consistency based MERGE mode early decision for HEVC," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 198–213, Apr. 2017.

[9] S.-H. Jung and H. W. Park, "A fast mode decision method in HEVC using adaptive ordering of modes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 10, pp. 1846–1858, Oct. 2016.

[10] I. Zupancic, S. G. Blasi, E. Peixoto, and E. Izquierdo, "Inter-prediction optimizations for video coding using adaptive coding unit visiting order," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1677–1690, Sep. 2016.

[11] X. Wu, H. Wang, and Z. Wei, "Bayesian rule based fast TU depth decision algorithm for high efficiency video coding," in *Proc. Vis. Commun. Image Process. (VCIP)*, Chengdu, China, Nov. 2016, pp. 1–4.

[12] Z. Pan, P. Jin, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast reference frame selection based on content similarity for low complexity HEVC encoder," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 516–524, Oct. 2016.

[13] J.-H. Lee, B.-G. Kim, D.-S. Jun, S.-H. Jung, and J. S. Choi, "Complexity reduction algorithm for prediction unit decision process in high efficiency video coding," *IET Image Process.*, vol. 10, no. 1, pp. 53–60, Jan. 2016.

[14] F. Chen, P. Li, Z. Peng, G. Jiang, M. Yu, and F. Shao, "A fast inter coding algorithm for HEVC based on texture and motion quad-tree models," *Signal Process., Image Commun.*, vol. 47, pp. 271–279, Sep. 2016.

[15] X. Shang, G. Wang, T. Fan, and Y. Li, "Statistical and spatiotemporal correlation based low-complexity video coding for high-efficiency video coding," *J. Electron. Imag.*, vol. 24, no. 2, Mar. 2015, Art. no. 023006.

[16] Y. Lu, H. Liu, Y. Lin, L. Shen, and H. Yin, "Efficient coding mode and partition decision for screen content intra coding," *Signal Process., Image Commun.*, vol. 68, pp. 249–257, Oct. 2018.

[17] L. Shen, Z. Zhang, and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1709–1722, Oct. 2014.

[18] Y. Zhang, S. Kwong, X. Wang, H. Yuan, Z. Pan, and L. Xu, "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2225–2238, Jul. 2015.

[19] D. Lee and J. Jeong, "Fast CU size decision algorithm using machine learning for HEVC intra coding," *Signal Process., Image Commun.*, vol. 62, pp. 33–41, Mar. 2018.

[20] G. Correa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC encoding decisions using data mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 660–673, Apr. 2015.

[21] N. Li, Y. Zhang, L. Zhu, W. Luo, and S. Kwong, "Reinforcement learning based coding unit early termination algorithm for high efficiency video coding," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 276–286, Apr. 2019.

[22] M. Hassan and T. Shanableh, "Predicting split decisions of coding units in HEVC video compression using machine learning techniques," *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 32735–32754, Nov. 2018.

[23] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, and Z. Peng, "Binary and multi-class learning based low complexity optimization for HEVC encoding," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 547–561, Sep. 2017.

[24] L. Zhu, Y. Zhang, S. Kwong, X. Wang, and T. Zhao, "Fuzzy SVM-based coding unit decision in HEVC," *IEEE Trans. Broadcast.*, vol. 64, no. 3, pp. 681–694, Sep. 2018.

[25] Y. Lu, H. Liu, Y. Lin, Y. Yao, and H. Yin, "Fast intra coding based on online learning for high efficiency video coding," *Optik*, vol. 167, pp. 136–143, Aug. 2018.

[26] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005.

[27] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. Keng Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[28] S. B. Gelfand, C. S. Ravishankar, and E. J. Delp, "An iterative growing and pruning algorithm for classification tree design," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 2, pp. 163–174, Nov. 1991.

[29] F. Bossen, *Common HM Test Conditions and Software Reference Configurations*, document JCT-VC, L1100, Geneva, Switzerland, 2013.

[30] G. Bjöntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, ITU-T SG16/Q6 VCEG Meeting, Austin, TX, USA, Apr. 2001.

YU LU (Member, IEEE) received the Ph.D. degree in communication and information system from Shanghai University, China, in 2009. Afterwards, he joined Hangzhou Dianzi University, and he is currently a Visiting Professor with Oregon State University. From 2013 to 2014, he served as a Research Scholar with Tongji University, Shanghai, China. He has contributed dozens of academic articles and patents in the fields of image processing and video compression. He was a recipient of Excellent Thesis Awards from the International Conference on Electrical and Electronics Engineering, in 2012.

XUDONG HUANG received the B.S. degree in communication engineering from Hangzhou Dianzi University, China, in 2017, where he is currently pursuing the M.S. degree. His research interest is video coding.

HUAPING LIU (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, in 1997. He has been a Full Professor with the School of Electrical Engineering and Computer Science, Oregon State University, since 2011. He has authored or coauthored over 100 refereed technical articles in international journals and conferences. He has served as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the IEEE COMMUNICATIONS LETTERS, and an Editor for the *Journal of Communications and Networks* and *Security and Communication Networks*.

YANG ZHOU received the Ph.D. degree in electronics and information engineering from Ningbo University, China, in 2014. He was a Visiting Professor with the University of Missouri, Kansas City, USA, from 2018 to 2019. He is currently an Associate Professor with Hangzhou Dianzi University. His research interests include image analysis and 3D video compression.

HAIBING YIN received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006. He is currently a Full Professor with Hangzhou Dianzi University, China. He was a Visiting Professor with the University of Waterloo, Canada, in 2014 and 2017, respectively. His research interest includes image and video processing.

LIQUAN SHEN received the Ph.D. degree in communication and information systems from Shanghai University, China, in 2008. He was a Visiting Professor with the University of Florida, from 2013 to 2014. Since 2008, he has been with the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He has authored or coauthored over 100 refereed technical articles in international journals and conferences in the fields of video coding and image processing. His research interests include high efficiency video coding, perceptual coding, video codec optimization, 3D TV, and video quality assessment.