

# Fuzzy SVM-Based Coding Unit Decision in HEVC

Linwei Zhu<sup>1</sup>, *Student Member, IEEE*, Yun Zhang<sup>2</sup>, *Senior Member, IEEE*, Sam Kwong<sup>3</sup>, *Fellow, IEEE*,  
Xu Wang, *Member, IEEE*, and Tiesong Zhao, *Member, IEEE*

**Abstract**—The latest video compression standard, High Efficiency Video Coding (HEVC), has greatly improved the coding efficiency compared to the predecessor H.264/AVC. However, equipped with the quadtree structure of coding tree unit partition and other sophisticated coding tools, HEVC brings a significant increase in the computational complexity. To address this issue, a coding unit (CU) decision method based on fuzzy support vector machine (SVM) is proposed for rate-distortion-complexity (RDC) optimization, where the process of CU decision is formulated as a cascaded multi-level classification task. The optimal feature set is selected according to a defined misclassification cost and a risk area is introduced for an uncertain classification output. To further improve the RDC performance, different regulation parameters in SVM are adopted and outliers in training samples are eliminated. Additionally, the proposed CU decision method is incorporated into a joint RDC optimization framework, where the width of risk area is adaptively adjusted to allocate flexible computational complexity to different CUs, aiming at minimizing computational complexity under a configurable constraint in terms of RD performance degradation. Experimental results show that the proposed approach can reduce 58.9% and 55.3% computational complexity on average with the values of Bjønteggard delta peak-signal-to-noise ratio as  $-0.075$  dB and  $-0.085$  dB and the values of Bjøntegaard delta bit rate as 2.859% and 2.671% under low delay  $P$  and random access configurations, respectively, which has outperformed the state-of-the-art fast algorithms based on statistical information and machine learning.

**Index Terms**—Misclassification cost, fuzzy support vector machine, coding unit decision, rate-distortion-complexity optimization, High Efficiency Video Coding.

## I. INTRODUCTION

RECENTLY, a large number of video clips are posted to website and social media platform, which greatly change our lives. However, the capability of data storage and transmission has been challenged by the dramatically increasing multimedia data. Therefore, an efficient video compression algorithm is supposed to be developed. Since 2010, the High Efficiency Video Coding (HEVC) [1] standard was calling for proposal and the first version was finalized in 2013. Compared with the predecessor H.264/AVC [2], HEVC can save about 50% bit rate with the same visual quality [1], which has a great benefit for video storage and transmission. However, it adopts the strategy of achieving Rate Distortion (RD) performance gain at the cost of computational complexity. In other words, the best coding parameter is eventually determined by RD cost calculation and comparison from all candidates. Therefore, the computational complexity limits its real-time applications.

In general, there are two research directions on the Rate-Distortion-Complexity (RDC) optimization for HEVC. One is computational complexity control, and the other is fast algorithm. The former aims to minimize the coding distortion under a given computational complexity constraint, while the latter is to minimize the computational complexity under a given RD performance degradation constraint. For computational complexity control, the key point is to establish the relationship between coding parameter and computational complexity. For example, Corrêa *et al.* [3] had investigated the relationship based on extensive experiments. Then the appropriate coding parameters can be easily selected by looking up table under a given computational complexity constraint. Deng *et al.* [4] presented a novel computational complexity control approach, in which a visual attention model was introduced to determine the range of depth for Coding Tree Unit (CTU) by the established relationship with computational complexity. For fast algorithm, it is to predict or estimate coding parameters directly without RD cost calculation, which can avoid checking all coding parameter candidates. Many modules in video coding have been optimized, such as Coding Unit (CU)/Prediction Unit (PU)/Transform Unit (TU) [5] mode determination [6], Motion Estimation (ME) [7] and reference frame selection [8]. Some of coding parameters are skipped due to being rarely used in the previous frames, and some of searching processes are early terminated, then the time can be saved. This paper concentrates on the fast algorithm.

Manuscript received June 16, 2017; revised September 11, 2017; accepted September 19, 2017. Date of publication November 8, 2017; date of current version August 8, 2018. This work was supported in part by the Natural Science Foundation of China under Grant 61672443, Grant 61671152, Grant 61501299, and Grant 61471348, in part by the Hong Kong RGC General Research Fund under Grant 9042322 (CityU 11200116) and Grant 9042489 (CityU 11206317), in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2016A030306022, in part by the Project for Shenzhen Science and Technology Development under Grant JSGG20160229202345378, and in part by the Shenzhen International Collaborative Research Project under Grant GJHZ20170314155404913. (Corresponding author: Sam Kwong.)

L. Zhu and S. Kwong are with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the City University of Hong Kong Shenzhen Institute, Shenzhen 518057, China (e-mail: lwzhu2-c@my.cityu.edu.hk; cssamk@cityu.edu.hk).

Y. Zhang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: yun.zhang@siat.ac.cn).

X. Wang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: wangxu@szu.edu.cn).

T. Zhao is with the College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China (e-mail: t.zhao@fzu.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TBC.2017.2762470

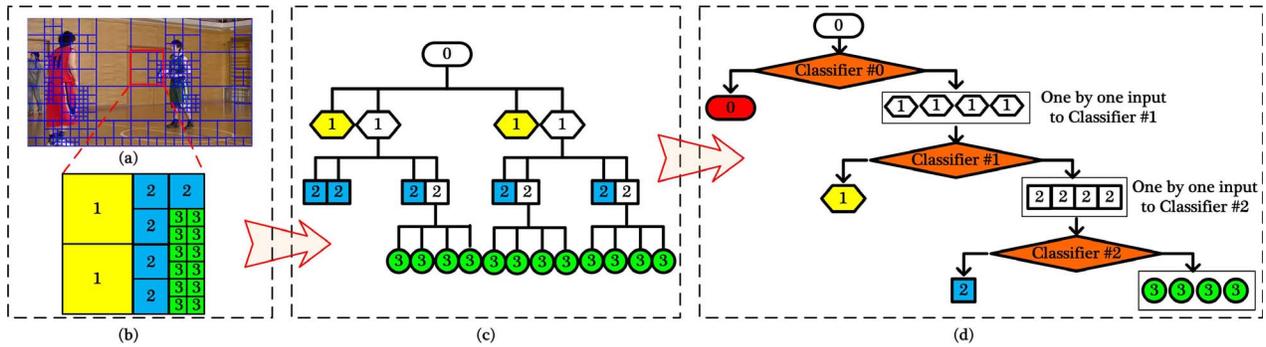


Fig. 1. CTU partitions in HEVC. (a) CU partition in a frame. (b) A CTU partition. (c) Quadtree structure of CTU partition. (d) Cascaded classifiers.

In this paper, we present a novel CU decision scheme based on Machine Learning (ML). The main technical contributions of this paper are listed as follows: (1) the optimal feature set is selected based on a defined misclassification cost; (2) regulation parameters in Support Vector Machine (SVM) [9] are re-determined due to the difference between False Positive (FP) and False Negative (FN) rates; (3) different weights are set for training samples to eliminate the negative influence from outliers; (4) the risk area is introduced to achieve a good trade-off between computational complexity reduction and RD performance degradation; (5) for different blocks, the flexible computational complexity is adaptively allocated by the variable widths of risk area.

The remainder of this paper is organized as follows. Section II briefly introduces the related works on fast algorithm and Section III presents the problems and motivations. The fuzzy SVM based CU decision is proposed in Section IV and the optimal width of risk area determination is discussed in Section V. Experimental results are illustrated in Section VI. Section VII draws the conclusions.

## II. OVERVIEW OF THE RELATED WORKS

To effectively reduce the computational complexity in HEVC, many researchers have done lots of works. Generally, the fast algorithms can be divided into two categories: statistical information based [10]–[20] and ML based [21]–[30].

Shen *et al.* [10], [11] presented a fast inter frame mode decision algorithm for HEVC, in which all existing correlations, including quadtree structured and spatial-temporal correlations, were jointly used to determine the range of block size and to skip some special levels that were rarely used in neighboring blocks. In [12], the coding parameters, such as Motion Vector (MV), TU size, and Coded Block Flag (CBF), were all utilized to estimate the texture complexity for a fast encoding scheme. Xiong *et al.* [13] proposed a fast inter CU decision scheme, in which a novel ME method was designed and a new concept named motion compensation RD cost was defined. To reduce the time of RD optimization, zero block detection scheme [14], [15] was adopted after Hadamard transform based on statistical information. In addition, a novel approach [16], bottom to top visiting order, was effectively developed to reduce the computational complexity. Jung and Park [17] proposed a fast mode decision method,

where the adaptive ordering of modes was employed. The skip mode was fast determined [18] using the distortion characteristics after calculating RD cost of  $2N \times 2N$  merge mode. Similarly, an early skip mode decision algorithm was presented in [19], where a unimodal stopping model was designed. The relationship between impossible modes and distribution of distortion was explored to accelerate the encoding by [20]. Basically, these aforementioned fast algorithms merely use hard thresholds from statistical properties and spatial-temporal correlations. Although they can reduce computational complexity in some degrees, they may have a risk of bringing undesirable performance to some special cases because of statistical thresholds.

A weighted SVM was utilized to predict CU early termination in [21] for computational complexity optimization. A data mining based fast HEVC encoding method was proposed [22], where ML algorithm of decision tree was used to predict early termination and then the procedures of CU, PU and TU determination were all optimized. Zhang *et al.* [23] modeled the quadtree structure of CTU partition as a three-level decision task, and two three-output classifiers were trained. In [24], a Bayesian decision rule based early termination method was reported, in which on-line learning and off-line learning were jointly applied to generate model parameters of classifiers. Similar to [24], the Neyman-Pearson based rule was utilized for fast mode decision algorithm [25]. A binary and multi-class SVM based fast HEVC encoding algorithm was presented in [26], where the off-line and on-line ML modes were combined for classifiers based on a multiple reviewers system. For the fast mode decision in intra frame encoding, the ML technique is utilized as well. For example, two SVMs that adopted the difference of CU sizes and RD cost ratio as features were proposed to perform the decisions of CU splitting and early CU termination in [27]. Duanmu *et al.* [28] proposed a fast mode and partition decision framework for Screen Content Coding (SCC), where several decision trees were trained and employed. Zhang *et al.* [29] presented an effective data driven CU size decision approach, in which two stages were included via off-line learning and on-line learning. In addition, a Convolutional Neural Network (CNN) based approach was devised to reduce the encoder's hardware complexity [30]. Generally, these schemes are all fast video coding methods based on ML. However, there will be a penalty of RD performance degradation in case of false classification

prediction. Therefore, an efficient misclassification prediction control mechanism is supposed to be developed.

Actually, the statistical information based fast algorithms have achieved a good performance. But they reduce computational complexity by removing rarely used modes according to the statistical distributions and thresholds. These statistical distributions and thresholds are different from sequence to sequence, which may lead to performance fluctuation, and it always has a penalty of RD performance degradation because of removing some candidates. While for ML based scheme, it does the computational complexity reduction by prediction, all modes are candidates. The performance can be improved if the sophisticated ML tools, appropriate features and optimal parameters are adopted. Therefore, the ML based algorithm has more potential to reduce computational complexity.

### III. PROBLEMS AND MOTIVATIONS

To achieve a better coding efficiency, the recursive CU decision has been introduced in HEVC. The CUs can change from  $64 \times 64$  to  $8 \times 8$ , and they are noted as Depth 0 to Depth 3. Fig. 1(a) illustrates an example of a frame from BasketballPass ( $416 \times 240$ ) sequence and Fig. 1(b) provides the detailed partitions of the CTU marked with red boundary in Fig. 1(a). The digital numbers in the blocks indicate the levels of depth. According to the rule of recursive CU decision, the current CU can be split into four sub-CUs, and then every sub-CU can be further split into four CUs in the subsequent depth until the largest level of depth. Eventually, the best CUs combination will be determined after RD cost comparison. The procedure demonstrates that every possible CU is required to be taken as a candidate for achieving the optimal combination. As such, the computational complexity of HEVC is about several times than that of H.264/AVC [1].

Fig. 1(c) shows the quadtree structure of Fig. 1(b), where different symbols indicate different CUs. This is a pruned quadtree since the unused nodes and leaves are removed accordingly. Compared with full quadtree, it only has limited nodes and leaves. The quadtree structure of CTU partitions (Fig. 1(c)) can be modeled as a cascaded classification task [23], as shown in Fig. 1(d). The task reveals that the splitting or non-splitting of current CU can be determined by the Classifiers #0, #1 and #2 illustrated in Fig. 1(d). With these cascaded classifiers, the full quadtree can be effectively pruned. In other words, some CUs are early terminated and some PU modes are skipped, then the computational complexity can be reduced.

To analyze the upper bound of computational complexity reduction with ML based CU decision scheme, four sequences including BasketballPass ( $416 \times 240$ ), BQMall ( $832 \times 480$ ), Johnny ( $1280 \times 720$ ) and Kimono1 ( $1920 \times 1080$ ) are adopted for information collocation. For each sequence, twenty one frames are encoded twice by the HEVC test model. The difference is that ground truths (CUs with the minimum RD cost from the first encoding) are utilized directly in the second encoding. The computational complexities of them are compared. Table I shows the experimental results. QP denotes the Quantization Parameter. D0 to D3 indicate Depth 0 to Depth 3,

TABLE I  
DEPTH DISTRIBUTION AND UPPER BOUND OF COMPUTATIONAL COMPLEXITY REDUCTION UNDER ML BASED CODING [UNIT: %]

| Sequence                    | QP | Depth Distribution |             |             |             | TS          |
|-----------------------------|----|--------------------|-------------|-------------|-------------|-------------|
|                             |    | D0                 | D1          | D2          | D3          |             |
| BasketballPass<br>(416×240) | 22 | 9.50               | 30.9        | 33.2        | 26.3        | 55.2        |
|                             | 27 | 12.1               | 33.2        | 34.7        | 18.9        | 59.4        |
|                             | 32 | 14.9               | 36.9        | 34.6        | 13.4        | 63.0        |
|                             | 37 | 18.3               | 41.0        | 32.2        | 8.39        | 65.8        |
| BQMall<br>(832×480)         | 22 | 18.2               | 36.4        | 28.5        | 16.7        | 55.9        |
|                             | 27 | 28.0               | 37.8        | 23.1        | 11.0        | 60.4        |
|                             | 32 | 37.0               | 37.4        | 18.5        | 6.92        | 64.0        |
|                             | 37 | 44.7               | 36.8        | 14.3        | 4.09        | 67.0        |
| Johnny<br>(1280×720)        | 22 | 61.4               | 24.2        | 12.6        | 1.69        | 68.9        |
|                             | 27 | 72.1               | 18.9        | 8.30        | 0.66        | 73.2        |
|                             | 32 | 78.8               | 14.5        | 6.29        | 0.33        | 75.0        |
|                             | 37 | 83.1               | 11.3        | 5.36        | 0.16        | 76.1        |
| Kimono1<br>(1920×1080)      | 22 | 26.6               | 45.2        | 22.3        | 5.70        | 67.4        |
|                             | 27 | 37.8               | 41.8        | 16.6        | 3.71        | 68.3        |
|                             | 32 | 48.3               | 36.7        | 12.2        | 2.62        | 70.5        |
|                             | 37 | 58.6               | 31.0        | 8.33        | 1.96        | 72.2        |
| <b>AVERAGE</b>              |    | <b>40.5</b>        | <b>32.1</b> | <b>19.4</b> | <b>7.65</b> | <b>66.4</b> |

respectively. TS indicates Time Saving. From the results, it can be found that about 40.5%, 32.1%, 19.4% and 7.65% pixel regions select Depth 0, 1, 2 and 3 as their best CUs, respectively. If these CUs can be correctly predicted by classifiers, the computational complexity reduction can reach 66.4% on average without any penalty of RD performance degradation, which is the upper bound of computational complexity reduction. At the same time, it demonstrates that there is much potential to reduce computational complexity with ML based CU decision.

SVM is utilized as the classifier in Fig. 1(d) for its better classification prediction performance. Given a training set with  $l$  samples  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_i, y_i), \dots, (\mathbf{x}_l, y_l)\}$ ,  $\mathbf{x}_i \in \mathbb{R}^n$  is feature vector of the  $i^{\text{th}}$  sample,  $n$  is the feature dimension and  $y_i \in \{-1, +1\}$  is ground truth of the  $i^{\text{th}}$  sample. The SVM classification task can be derived as [9]:

$$\begin{aligned} \min & \left( \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l \xi_i \right) \\ \text{s.t.} & y_i (\omega^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, 2, \dots, l \end{aligned} \quad (1)$$

where  $\phi(\mathbf{x}_i)$  transfers  $\mathbf{x}_i$  from a lower-dimensional space into a higher-dimensional space and the regulation parameter  $C$  controls the balance between margin maximum and false classification prediction.  $(\omega, b)$  represent the hyperplane.  $\xi_i$  is slack variable of the  $i^{\text{th}}$  sample, which indicates the misclassification. The decision function is written as [9]:

$$y(\mathbf{x}) = \text{sgn}(\omega^T \phi(\mathbf{x}) + b) = \begin{cases} +1, & \omega^T \phi(\mathbf{x}) + b > 0 \\ -1, & \omega^T \phi(\mathbf{x}) + b \leq 0, \end{cases} \quad (2)$$

where  $\mathbf{x}$  is the feature vector of the current test sample.

Basically, although SVM is able to solve majority of real-world classification problems, there are still some issues which should be considered to improve the performance of ML based video coding. One issue is outlier. Some samples are far away from central sample, which may have negative influence to classifier training. As shown in Fig. 2, it is the histogram of the distance between the current and central samples. Three sequences, BasketballPass ( $416 \times 240$ ), BQMall

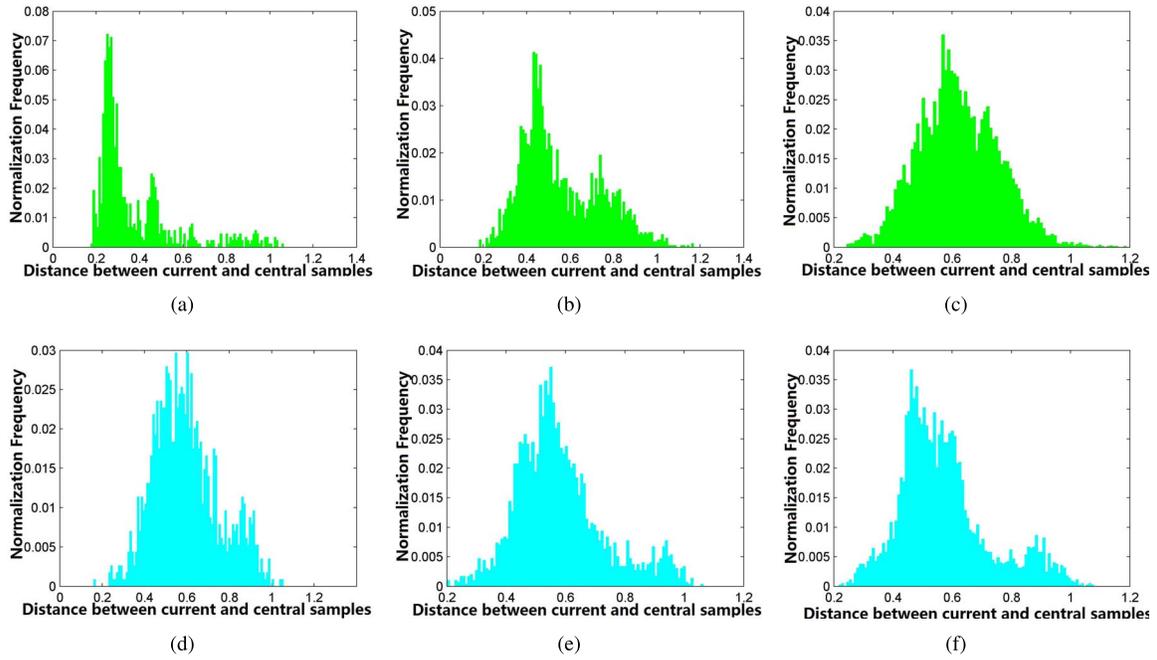


Fig. 2. Histogram of distance between the current and central samples. (a) Classifier #0 (Negative Sample Set). (b) Classifier #1 (Negative Sample Set). (c) Classifier #2 (Negative Sample Set). (d) Classifier #0 (Positive Sample Set). (e) Classifier #1 (Positive Sample Set). (f) Classifier #2 (Positive Sample Set).

(832×480) and Johnny (1280×720), are adopted for features [23] and ground truths (splitting (+1) and non-splitting (-1)) collection. The value of central sample is the average value of the samples in terms of feature vector. The distance is represented by Euclidean distance. It can be easily found that the distances of some samples are larger than those of others. Another issue is the misclassification cost. In real-world problems, the costs of the FP and FN rates are always different. They should be assigned different weights to minimize the false classification cost. However, these two issues are not seriously taken into consideration in these existing works [23], [26], [31]. Specifically, when the classification prediction accuracy (splitting/non-splitting) is low, the RD performance cannot be guaranteed by using the ML based structure directly. Thus, an efficient control mechanism is also supposed to be developed to achieve a better RDC performance.

#### IV. PROPOSED FUZZY SVM BASED CU DECISIONS

##### A. ML Based CU Decision

Based on the analysis in Section III, the splitting or non-splitting of the current CU is determined by a classifier who has learned the mapping from feature vector to CTU partitions. Beginning with a CTU, when Depth is less than 3, a classifier is utilized to predict the splitting flag. If the output is splitting, the current CU will be split into four sub-CUs directly; If the output is non-splitting, the PU candidates will be checked and the remaining CUs will be skipped; If the output is uncertain, the original HEVC test model will be activated.

Generally, in a SVM, if a test sample is far away from the hyperplane, it is more confident for its classification prediction [32]. Therefore, there is a risk area [32]. If a test sample

belongs to this risk area, there is a high risk of misclassification. The distance between test sample and hyperplane can be represented as:

$$d = \frac{\omega^T \phi(\mathbf{x}) + b}{\|\omega\|}, \quad (3)$$

where  $\phi(\mathbf{x})$  and  $(\omega, b)$  are as same as the parameters in Eq.(1). In order to achieve a good trade-off between computational complexity reduction and RD performance degradation, we redesign the decision function. If a test sample does not belong to the risk area, the output will be accepted; otherwise, the output will be rejected, and set as 0, which means uncertain output.

$$Y(\mathbf{x}) = \begin{cases} 0, & \beta^- \leq d - \delta \leq \beta^+ \\ y(\mathbf{x}), & \text{otherwise,} \end{cases} \quad (4)$$

where  $\delta$  is the offset of risk area with respect to the hyperplane. In this paper,  $\delta$  is set as 0 for simplicity.  $y(\mathbf{x})$  and  $d$  are determined by Eqs.(2) and (3), respectively.  $\beta^-$  and  $\beta^+$  are the lower and upper boundaries of risk area, which will be determined in Section V.

Some statistical experiments are conducted to reveal the influence of risk area in the classifier. Four sequences are selected, *i.e.*, BasketballPass (416×240), BQMall (832×480), KristenAndSara (1280×720), and Kimono1 (1920×1080). For simplicity, the parameters of  $\beta^-$  and  $\beta^+$  are set as -0.005 and +0.005. The classification prediction results are shown in Table II. It can be found that the total classification prediction accuracy could reach 81.7% on average, while the classification prediction accuracy in the risk area only reaches 27.6% on average. At the same time, the rates of FP and FN are presented as 4.44% and 13.9% respectively. It indicates that there is a large gap between FP and FN. Thus, the different misclassification costs from FP and FN are supposed to

TABLE II  
CU SPLITTING/NON-SPLITTING CLASSIFICATION PREDICTION ACCURACY [UNIT: %] (RATIO1 AND RATIO2 INDICATE TOTAL CLASSIFICATION PREDICTION ACCURACY AND CLASSIFICATION PREDICATION ACCURACY IN RISK AREA)

| Classifier     | Sequence       | Ratio1      | Ratio2      | FP Rate     | FN Rate     |
|----------------|----------------|-------------|-------------|-------------|-------------|
| #0             | BasketballPass | 97.2        | 0.00        | 1.39        | 1.39        |
|                | BQMall         | 87.6        | 0.00        | 2.47        | 9.89        |
|                | KristenAndSara | 87.9        | 50.0        | 3.75        | 8.29        |
|                | Kimono1        | 90.2        | 0.00        | 0.16        | 9.69        |
| #1             | BasketballPass | 92.3        | 0.00        | 1.64        | 6.15        |
|                | BQMall         | 82.0        | 50.0        | 3.58        | 14.4        |
|                | KristenAndSara | 75.1        | 39.5        | 5.79        | 19.2        |
|                | Kimono1        | 65.8        | 51.4        | 15.0        | 19.2        |
| #2             | BasketballPass | 74.7        | 51.2        | 1.67        | 23.6        |
|                | BQMall         | 62.4        | 0.00        | 1.85        | 35.8        |
|                | KristenAndSara | 78.5        | 51.7        | 3.57        | 17.9        |
|                | Kimono1        | 86.4        | 37.0        | 12.4        | 1.19        |
| <b>AVERAGE</b> |                | <b>81.7</b> | <b>27.6</b> | <b>4.44</b> | <b>13.9</b> |

TABLE III  
SELECTED FEATURES OF DIFFERENT CLASSIFIERS ( $X_{CBF\_SKIP}$  IS FROM BY-PRODUCT INFORMATION OF THE CURRENT CU CODING, WHILE  $X_{CBF\_NB}$  IS FROM THE SPATIAL DOMAIN)

| Index | Candidates        | Selected Features |               |               |
|-------|-------------------|-------------------|---------------|---------------|
|       |                   | Classifier #0     | Classifier #1 | Classifier #2 |
| 1     | $x_{SAD}$         | ✓                 |               | ✓             |
| 2     | $x_{RDCost}$      | ✓                 | ✓             | ✓             |
| 3     | $x_{SkipFlag}$    |                   | ✓             | ✓             |
| 4     | $x_{Distortion}$  | ✓                 | ✓             |               |
| 5     | $x_{Bits}$        |                   | ✓             | ✓             |
| 6     | $x_{CtxSkipFlag}$ |                   |               | ✓             |
| 7     | $x_{QP}$          | ✓                 |               | ✓             |
| 8     | $x_{CBF\_SKIP}$   | ✓                 |               | ✓             |
| 9     | $x_{MergeFlag}$   |                   |               | ✓             |
| 10    | $x_{MV}$          |                   |               | ✓             |
| 11    | $x_{Partition}$   |                   |               | ✓             |
| 12    | $x_{Depth}$       | ✓                 |               | ✓             |
| 13    | $x_{CBF\_NB}$     | ✓                 |               | ✓             |

be considered in the ML based video coding, which will be discussed in Section IV-C.

### B. Feature Selection

As we know, features are important for classifiers, since they have significant influence on the classification prediction accuracy. Therefore, the optimal feature set is desired to achieve a better performance. As shown in Table III, thirteen feature candidates from the temporal domain, by-product information of the current CU coding and the spatial domain are listed. The detailed descriptions of these feature candidates are presented as follows.

The first category of features are extracted from the temporal domain. The value of Sum of Absolute Difference (SAD) between current and co-located (in position 0 of the reference list 0) CUs is calculated as (1)  $x_{SAD}$ .

The second category of features are extracted from the by-product information of the current CU coding. The current coding information after checking SKIP mode is highly valuable, which is worth setting as features. Therefore, the corresponding by-product information is extracted and denoted as (2)  $x_{RDCost}$ , (3)  $x_{SkipFlag}$ , (4)  $x_{Distortion}$ , (5)  $x_{Bits}$ , (6)  $x_{CtxSkipFlag}$ . The RD cost, distortion, and bits indicate the RD performance. Thus they are listed as features. SkipFlag is the flag of skipping in PU level. CtxSkipFlag is the flag of skipping in neighboring

blocks. Additionally, QP, the adjusting factor between coding bits and distortion, is introduced as the feature (7)  $x_{QP}$  since it will affect the CTU partitions. (8)  $x_{CBF\_SKIP}$ , CBF is an important flag representing coded block, which relates to the coding information.

The third category of features are extracted from the spatial domain. Five features in adjacent encoded blocks are selected and denoted as (9)  $x_{MergeFlag}$ , (10)  $x_{MV}$ , (11)  $x_{Partition}$ , (12)  $x_{Depth}$ , and (13)  $x_{CBF\_NB}$ , respectively. MergeFlag denotes the flag of merge mode. MV is the motion information. Partition size means the selected PU size from mode candidates. CU depth is the block size, which can change from  $64 \times 64$  to  $8 \times 8$ . And the coded block flags from neighboring blocks are extracted. These five features are from upper-above, left, upper-right blocks of current block, and the value of individual feature is calculated by weighted averaging.

In the existing ML based video coding schemes, the features are mostly selected by experience. Generally, the performance can be further improved with an optimal feature set. In [31], a feature selection approach was proposed via checking all the feature set candidates. However, only the classification prediction accuracy was set as the cost function. For the ML based video coding, the RD performance should be taken into account for feature selection. Then the proposed cost function of feature selection is designed with the RD performance degradation due to misclassification.

$$J = \mu^+ P_{fp} + \mu^- P_{fn}, \quad (5)$$

where  $\mu^+$  and  $\mu^-$  are the RD cost ratios with false classification prediction,  $P_{fp}$  and  $P_{fn}$  are the FP and FN rates, respectively. Here,  $\mu^+$  and  $\mu^-$  can be calculated as follows:

$$\mu^+ = \frac{RDCost_s}{RDCost_{best}}, \mu^- = \frac{RDCost_{ns}}{RDCost_{best}}, \quad (6)$$

where  $RDCost_{ns}$  is the RD cost when the current block is not split into four sub-CUs in case of false negative prediction,  $RDCost_s$  is the RD cost when the current block is split into four sub-CUs in case of false positive prediction, and  $RDCost_{best}$  indicates the RD cost when the current block achieves the optimal choice. For three classifier levels, the parameters of  $\mu^+$  and  $\mu^-$  are set as 1.185 : 1.014, 1.089 : 1.038 and 1.026 : 1.057, respectively, according to [23].

Based on the principle of checking all feature set candidates, the total number of feature sets is calculated by:

$$K = \sum_{i=1}^N \binom{i}{N} = 2^N - 1, \quad (7)$$

where  $N$  is the number of feature candidates. Here,  $N$  equals to 13. So the total number of feature sets is  $K = 8191$ , and the numbers from 1 to 8191 are assigned to the feature sets as indexes. For example, feature set #1 contains {feature #1}, ..., feature set #14 contains {feature #1, feature #2}, etc.

The misclassification costs of all these feature sets are calculated, as shown in Fig. 3. The red lines indicate the feature sets with minimum misclassification cost. Therefore, feature sets #5065, #598 and #8187 are optimal for Classifiers #0,

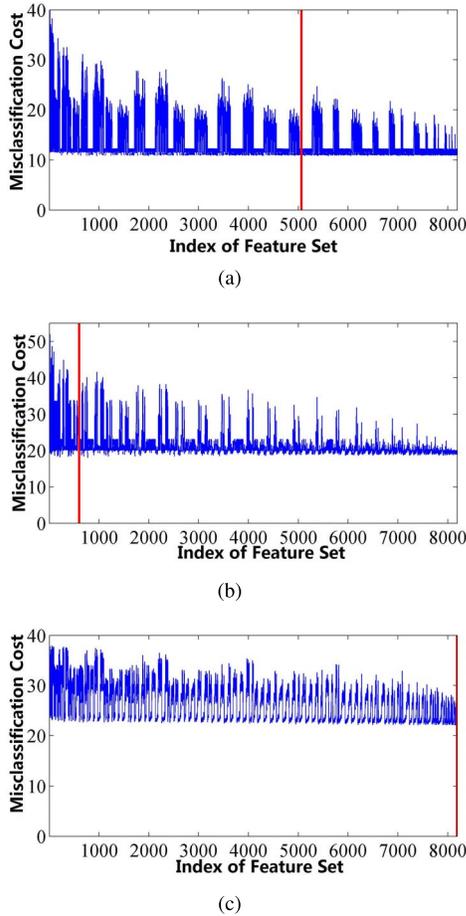


Fig. 3. Misclassification cost of feature set. (a) Classifier #0. (b) Classifier #1. (c) Classifier #2. (Red lines indicate feature sets with minimum misclassification cost, and their indexes are 5065, 598 and 8187, respectively.)

#1 and #2. Then, the selected features are achieved, which are shown in right columns in Table III. It can be found that (1) for different classifiers, the optimal feature sets are different; (2) if all these feature candidates are utilized, it does not always guarantee the best performance.

### C. Regulation Parameters Determination for SVM

In the real-world classification problems, the costs of FP and FN rates are always different. However, the regulation parameters in Eq.(1) are identical. In this paper, for the CTU partitions, the splitting prediction is defined as positive (+1), and the non-splitting prediction is negative (-1). The regulation parameters of these two classes are noted as  $C^+$  and  $C^-$ . Then Eq.(1) becomes as follows:

$$\begin{aligned} \min & \left( \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l \xi_i \right), \\ C & = \begin{cases} C^+, & y_i > 0 \\ C^-, & y_i < 0, \end{cases} \\ \text{s.t.} & \quad y_i (\omega^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \quad \xi_i \geq 0, i = 1, 2, \dots, l. \end{aligned} \quad (8)$$

To determine the optimal regulation parameters of  $C^+$  and  $C^-$ , four sequences are utilized, *i.e.*, BasketballPass

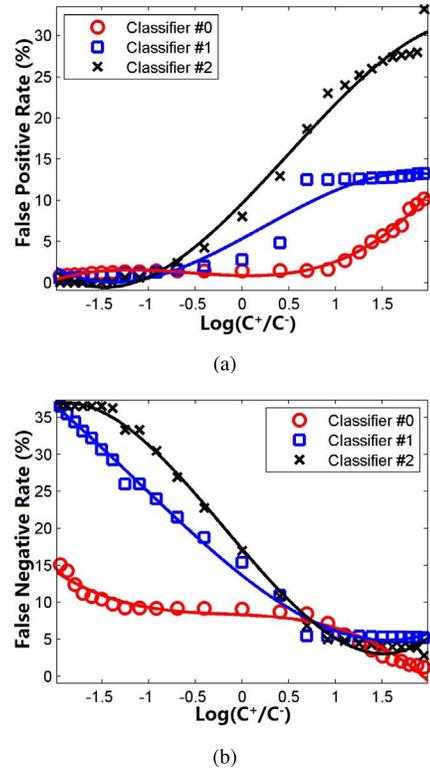


Fig. 4. False rate with respect to  $\log(C^+/C^-)$ . (a) FP vs  $\varphi$ . (b) FN vs  $\varphi$ .

TABLE IV  
FITTING PARAMETERS AND FITTING ACCURACY

| Functions                                 | Parameter | Classifier Level |               |               |
|---|-----------|------------------|---------------|---------------|
|   |           | #0               | #1            | #2            |
| $p(\varphi) = \sum_{i=0}^3 A_i \varphi^i$ | $A_3$     | 0.708            | -0.8253       | -1.07         |
|   | $A_2$     | 1.185            | 0.5408        | 1.584         |
|   | $A_1$     | -0.187           | 6.148         | 11.58         |
|   | $A_0$     | 0.8428           | 5.33          | 9.615         |
|   | $R^2$     | <b>0.9893</b>    | <b>0.9590</b> | <b>0.9916</b> |
| $q(\varphi) = \sum_{i=0}^3 B_i \varphi^i$ | $B_3$     | -0.7815          | 0.5597        | 1.718         |
|   | $B_2$     | -0.394           | 1.888         | 0.9668        |
|   | $B_1$     | -0.6012          | -9.924        | -14.77        |
|   | $B_0$     | 8.303            | 13.62         | 17.22         |
|   | $R^2$     | <b>0.9619</b>    | <b>0.9953</b> | <b>0.9970</b> |

(416 × 240), BQMall (832 × 480), Johnny (1280 × 720), and ParkScene (1920 × 1080). We define the relationship firstly:

$$\varphi = \log\left(\frac{C^+}{C^-}\right), \quad (9)$$

where  $\log()$  is the logarithmic function.

With the samples (features and ground truth) from these sequences, the classification prediction experiments are carried out. In Fig. 4, the FP rate increases while the FN rate decreases with the increasing  $\varphi$ . Then they are fitted by a polynomial function, and the fitting parameters and fitting accuracy are illustrated in Table IV.  $p(\varphi)$  and  $q(\varphi)$  are polynomial functions to represent the FP and FN rates with respect to  $\varphi$ , respectively.  $R^2$  means the fitting accuracy to evaluate the goodness of fitting, which ranges from 0 to 1. The larger value of  $R^2$  is, the better fitting is and vice versa. Here, for the two dimensional function fitting, the values of  $R^2$  are all

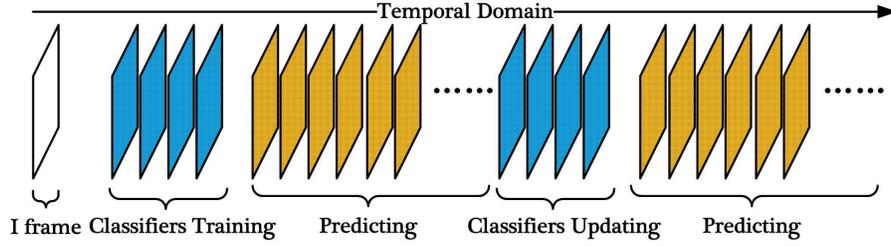


Fig. 5. Classifiers training and updating for the proposed method. (Note: classifiers are periodically updated during encoding.)

greater than 0.95, which indicate perfect fitting.  $A_i$  and  $B_i$  are the fitting parameters.

$$p(\varphi) = \sum_{i=0}^3 A_i \varphi^i, \quad q(\varphi) = \sum_{i=0}^3 B_i \varphi^i. \quad (10)$$

Then the optimal parameter of  $\varphi$  can be achieved by minimizing the misclassification cost.

$$\varphi^* = \arg \min_{\varphi} J, \quad (11)$$

where,

$$J = \mu^+ p(\varphi) + \mu^- q(\varphi). \quad (12)$$

After solving the above equation,

$$\varphi^* = \frac{\sqrt{\beta^2 - 4\alpha\gamma} - \beta}{2\alpha}. \quad (13)$$

Here, the parameters of  $\alpha$ ,  $\beta$  and  $\gamma$  are derived by  $A_1$  to  $A_3$ ,  $B_1$  to  $B_3$  and  $\mu^+$ ,  $\mu^-$ :

$$\begin{aligned} \alpha &= 3A_3\mu^+ + 3B_3\mu^-, \\ \beta &= 2A_2\mu^+ + 2B_2\mu^-, \\ \gamma &= A_1\mu^+ + B_1\mu^-. \end{aligned} \quad (14)$$

Therefore, the relationship of  $C^+$  and  $C^-$  in Eq.(9) can be rewritten as

$$C^+ = \exp(\varphi^*) \times C^-, \quad (15)$$

where  $\exp()$  is the exponential function. In this paper, we set  $C^-$  as 1.0, so  $C^+$  equals to  $\exp(\varphi^*)$ .

#### D. Fuzzy SVM Based Sample Weight Assignment

During the process of traditional classifier training, thousands of samples are treated as the same. Actually, they have different contributions. Some of them are useful and others may have negative influence. Therefore, they are supposed to be assigned different weights according to their individual contributions. Then, fuzzy SVM [33] is adopted, and Eq.(1) can be improved:

$$\begin{aligned} \min & \left( \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l k_i \xi_i \right), \\ \text{s.t.} & \quad y_i (\omega^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \quad \xi_i \geq 0, i = 1, 2, \dots, l, \end{aligned} \quad (16)$$

where  $k_i$  is the weight of the  $i^{\text{th}}$  sample. If a sample is far away from the central sample in the same class, it can be defined as

an outlier, which should be assigned a small weight. In other words, the slack variable of the sample will be ignored when training. Thus, the weight  $k_i$  of a sample can be used for slack variable in Eq.(16). Based on the distance between the current and central samples,  $k_i$  is calculated as follows:

$$k_i = \frac{2}{1 + \exp(\psi \|\mathbf{x}_i - \bar{\mathbf{x}}\|)}, \quad (17)$$

where  $\psi$  is a constant, which is set as 0.1 in this paper from extensive experiments,  $\mathbf{x}_i$  and  $\bar{\mathbf{x}}$  are the feature vectors of the current and central samples.

#### E. Overall Algorithm

According to Eqs.(1), (8) and (16), the above mentioned issues are all taken into account, then we achieve the following equation, which includes different regulation parameters ( $C^+$  and  $C^-$ ), and different weights  $k_i$  for samples.

$$\begin{aligned} \min & \left( \frac{1}{2} \omega^T \omega + C \sum_{i=1}^l k_i \xi_i \right), \\ C &= \begin{cases} C^+, y_i > 0 \\ C^-, y_i < 0, \end{cases} \\ \text{s.t.} & \quad y_i (\omega^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \quad \xi_i \geq 0, i = 1, 2, \dots, l. \end{aligned} \quad (18)$$

Here,  $C^+$  and  $C^-$  are determined by Eq.(15), and  $k_i$  is calculated by Eq.(17).

Based on the mentioned procedures, the overall algorithm of proposed method can be summarized as follows:

**Step 1:** As shown in Fig. 5, when a sequence is input for encoding, the frames of first Group of Picture (GOP) are encoded by the original HEVC test model. At the same time, the ground truths and the selected features are extracted;

**Step 2:** Classifiers #0, #1 and #2 are trained with the Eq.(18), and these classifiers can be periodically updated;

**Step 3:** After the classifiers are achieved, the CU partitions of following inter frames will be predicted directly;

**Step 4:** If a sample belongs to risk area, the original HEVC test model will be triggered, otherwise the outputs from classifiers will be utilized for encoding directly.

#### V. WIDTH OF RISK AREA DETERMINATION

According to Eq.(4), the width of risk area in the  $i^{\text{th}}$  CU level can be represented as

$$W_i = \beta_i^+ - \beta_i^-. \quad (19)$$

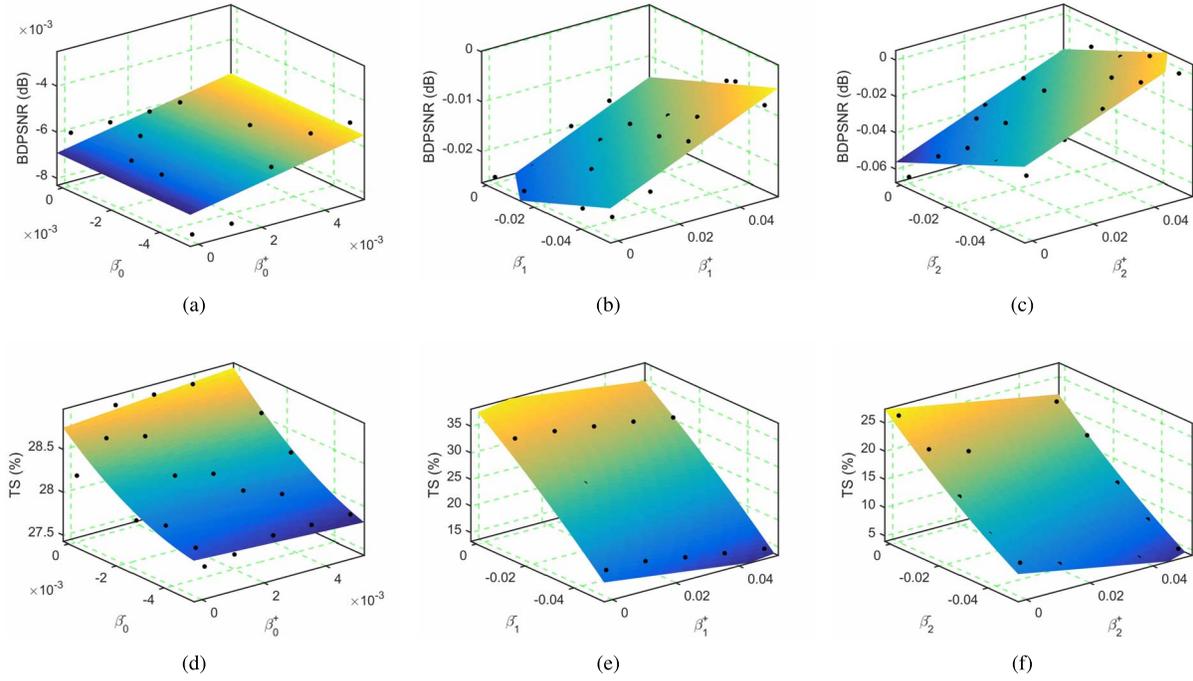


Fig. 6. Relationships among BD-PSNR, TS and parameters of  $\beta_i^-$  and  $\beta_i^+$ . (a) BD-PSNR vs  $\beta_0^-$  and  $\beta_0^+$  (Level 0). (b) BD-PSNR vs  $\beta_1^-$  and  $\beta_1^+$  (Level 1). (c) BD-PSNR vs  $\beta_2^-$  and  $\beta_2^+$  (Level 2). (d) TS vs  $\beta_0^-$  and  $\beta_0^+$  (Level 0). (e) TS vs  $\beta_1^-$  and  $\beta_1^+$  (Level 1). (f) TS vs  $\beta_2^-$  and  $\beta_2^+$  (Level 2).

In fact, there are several cases as follows:

- 1) When  $\beta_i^-$  and  $\beta_i^+$  equal to  $-\infty$  and  $+\infty$  respectively, all the outputs from classifier will be rejected, namely, the original HEVC test model is activated and there is no computational complexity optimization.
- 2) When  $\beta_i^-$  and  $\beta_i^+$  equal to  $-\infty$  and 0 respectively, the non-splitting classification prediction will always be rejected and the splitting classification prediction will always be accepted.
- 3) When  $\beta_i^-$  and  $\beta_i^+$  equal to 0 and  $+\infty$  respectively, the splitting classification prediction will always be rejected and the non-splitting classification prediction will always be accepted.
- 4) When  $-\infty \ll \beta_i^- < 0$  and  $0 < \beta_i^+ \ll +\infty$ , some of splitting classification predictions will be rejected and some of non-splitting classification predictions will be accepted.

To achieve the optimal parameters of  $\beta_i^-$  and  $\beta_i^+$ , statistical experiments are conducted firstly. Four sequences, *i.e.*, BasketballPass (416×240), BQMall (832×480), Johnny (1280×720), and ParkScene (1920×1080) are utilized for encoding. The parameters of  $\beta_i^-$  and  $\beta_i^+$  are set as ranging from 0 to -0.05 step by -0.0125 and from 0 to 0.05 step by 0.0125, respectively. The performances are measured by TS and Bjøntegaard Delta Peak-Signal-to-Noise Ratio (BD-PSNR) [34]. Fig. 6 shows the experimental results under different optimization levels. Level 0 represents that only Classifier #0 is activated and the Classifiers #1 and #2 are invalid. Similar configuration is set for Levels 1 and 2. It should be noted that if the classifier is invalid, the original HEVC test model will be utilized in the associated level. Here,

TABLE V  
FITTING PARAMETERS AND FITTING ACCURACY

| Functions                   | Parameter | Classifier Level |               |               |
|-----------------------------|-----------|------------------|---------------|---------------|
|                             |           | #0               | #1            | #2            |
| $r_i(\beta_i^-, \beta_i^+)$ | $a_i$     | 28.7             | 38.17         | 26.15         |
|                             | $b_i$     | 26.8             | -80.98        | -101.2        |
|                             | $c_i$     | 251              | 353           | 402.1         |
|                             | $d_i$     | 8320             | -44.8         | 23.5          |
|                             | $e_i$     | 19050            | -1035         | 13.4          |
|                             | $R^2$     | <b>0.8501</b>    | <b>0.9729</b> | <b>0.9868</b> |
| $p_i(\beta_i^-, \beta_i^+)$ | $f_i$     | -0.0068          | -0.02804      | -0.05316      |
|                             | $g_i$     | 0.2585           | 0.2696        | 0.6682        |
|                             | $h_i$     | 0.0057           | -0.1795       | -5588         |
|                             | $R^2$     | <b>0.8610</b>    | <b>0.9246</b> | <b>0.9060</b> |

TS is calculated as

$$TS = \frac{1}{4} \sum_{i=1}^4 \frac{T_c(QP_i) - T_\Psi(QP_i)}{T_c(QP_i)}, \quad (20)$$

where  $T_c(QP_i)$  and  $T_\Psi(QP_i)$  are the coding time of the original HEVC test model and scheme  $\Psi$  with  $QP_i$ , here  $\Psi$  indicates the proposed method.

We model the parameters ( $\beta_i^-$  and  $\beta_i^+$ ) determination as a RDC optimization issue, *i.e.*, maximizing the computational complexity reduction  $R$  and maintaining the value  $P$  of BD-PSNR being greater than a pre-defined threshold  $t$ :

$$\max[R], \text{ s.t. } P \geq t. \quad (21)$$

For simplicity,  $r_i(\beta_i^-, \beta_i^+)$  and  $p_i(\beta_i^-, \beta_i^+)$  are utilized to demonstrate the fitting functions of TS and BD-PSNR with respect to the parameters of  $\beta_i^-$  and  $\beta_i^+$ , and the fitting parameters and accuracy are shown in Table V,  $a_i$  to  $h_i$  are the fitting parameters,  $R^2$  means the fitting accuracy to measure

the goodness of fitting, which ranges from 0 to 1. For the three dimensional function fitting, the values of  $R^2$  are all greater than 0.85, which indicate good fitting. For different classifier levels, the fitting parameters are different.

$$r_i(\beta_i^-, \beta_i^+) = a_i + b_i\beta_i^- + c_i\beta_i^+ + d_i\beta_i^-\beta_i^+ + e_i(\beta_i^+)^2, \quad (22)$$

$$p_i(\beta_i^-, \beta_i^+) = f_i + g_i\beta_i^- + h_i\beta_i^+, \quad (23)$$

where  $a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i$  are the fitting parameters.

Then, the Eq.(21) can be rewritten as

$$\max \left[ \sum_{i=0}^2 m_i r_i(\beta_i^-, \beta_i^+) \right], \text{ s.t. } \sum_{i=0}^2 n_i p_i(\beta_i^-, \beta_i^+) \geq t, \quad (24)$$

where  $m_i$  and  $n_i$  are the weights, demonstrating the contribution to the ultimate BD-PSNR and TS. Generally, this optimization issue can be solved by the Lagrange Multiplier Method as follows:

$$(\beta_0^-, \beta_0^+, \beta_1^-, \beta_1^+, \beta_2^-, \beta_2^+)^* = \operatorname{argmax} J, \quad (25)$$

where,

$$J = \sum_{i=0}^2 m_i r_i(\beta_i^-, \beta_i^+) - \lambda \left( \sum_{i=0}^2 n_i p_i(\beta_i^-, \beta_i^+) - t \right), \quad (26)$$

where  $\lambda$  is the Lagrange Multiplier. Then, we can get the following linear equations,

$$\begin{cases} m_0(b_0 + d_0\beta_0^+) - \lambda n_0 g_0 = 0 \\ m_0(c_0 + d_0\beta_0^- + 2e_0\beta_0^+) - \lambda n_0 h_0 = 0 \\ m_1(b_1 + d_1\beta_1^+) - \lambda n_1 g_1 = 0 \\ m_1(c_1 + d_1\beta_1^- + 2e_1\beta_1^+) - \lambda n_1 h_1 = 0 \\ m_2(b_2 + d_2\beta_2^+) - \lambda n_2 g_2 = 0 \\ m_2(c_2 + d_2\beta_2^- + 2e_2\beta_2^+) - \lambda n_2 h_2 = 0 \\ \sum_{i=0}^2 n_i (f_i + g_i\beta_i^- + h_i\beta_i^+) = t. \end{cases} \quad (27)$$

After solving the above linear equations, the optimal parameters of  $(\beta_0^-, \beta_0^+, \beta_1^-, \beta_1^+, \beta_2^-, \beta_2^+)$  are achieved for three classifier levels. The pre-defined threshold  $t$  in Eq.(27) is set as -0.100 in terms of BD-PSNR. It should be noted that the width of risk area is only determined once when the RD performance degradation constraint is given before encoding.

## VI. EXPERIMENTAL RESULTS AND ANALYSIS

The experiments are conducted on the platform of HEVC test model version HM 16.5 [35]. Nineteen sequences from Class A to Class E are encoded at four QPs, 22, 27, 32, and 37 under Common Test Conditions (CTC) [36] with Low Delay P (LDP) and Random Access (RA) configurations. The settings of Fast Encoder Decision and Fast Decision for Merge RD cost are enabled for HM 16.5. The settings of Early SKIP Detection (ESD), Early CU (ECU), and CBF Fast Mode (CFM) are disabled, and the default configuration is used. All the experiments are performed on the computer equipped with the Intel Core i7-4790 CPU @ 3.60GHz, 8GB memory, Windows 7 Enterprise 64-bit operating system. The original HEVC test model (HM 16.5) is utilized as the anchor for performance comparison. The computational complexity reduction is measured by TS, the RD performance gain is measured by BD-PSNR and Bjøntegaard Delta Bit Rate (BD-BR) [34].

The larger value of TS means more computational complexity has been reduced, the negative value of BD-PSNR and positive value of BD-BR demonstrate RD performance degradation. The coding time is the total executing time of video encoder. For the proposed method, the time of classifiers training is also included, because the Classifiers #0, #1 and #2 are trained by on-line mode with training data from the first GOP and updated at a fixed period during encoding, as shown in Fig. 5. After the classifiers are prepared, the CTU partitions in the following inter frames are directly predicted. It should be noted that the sequences in Class A under LDP configuration and the sequences in Class E under RA configuration are not tested according to the CTC [36]. Since the sequences of BasketballPass, BQMall, Johnny and ParkScene have been used for fitting parameters determination, they are marked as \* just for reference in the results. In addition, the source codes of proposed feature selection and ML based CU decisions are available for academic use.<sup>1</sup>

### A. Performance Evaluation of Feature Selection

Firstly, the proposed approach with full features and the proposed approach with selected features under three optimization levels are compared to evaluate the efficient of feature selection. Also, the results of combined optimization levels are provided. For a fair performance comparison, the widths of risk area are set as the same, *i.e.*,  $\beta^- = -0.005$  and  $\beta^+ = +0.005$ . And the classifiers are all trained by Eq.(1). The experimental results are shown in Tables VI and VII. Level 0 + 1 + 2 means that Classifiers #0, #1 and #2 are all activated, which demonstrates that these three separate optimization levels are combined together.

Although only 4 features are selected for Level 1 (shown in Table III), according to the results in Tables VI and VII, the performances of proposed approach with selected features are better than those of proposed approach with full features, where the BD-BR of the former is 0.950% while the BD-BR of the latter is 1.128%, the BD-PSNR of the former is -0.023dB while the BD-PSNR of the latter is -0.027dB, and the TS of the former is 37.7% while the TS of the latter is 36.3%. Similar results can be found in other levels. It demonstrates that the proposed approach with selected features outperforms the proposed approach with full features. Meanwhile, it can be found that Level 1 has the largest computational complexity reduction while Level 0 keeps the best RD performance.

### B. Performance Comparison With State-of-the-Art Algorithms

In this subsection, four state-of-the-art fast HEVC encoding algorithms, including Corrêa CSVT [22], Zhang TIP [23], Zupancic TMM [16], and Jung CSVT [17], are utilized as benchmarks for performance comparison. The first two are ML based schemes while the other two are the statistical information based schemes. The source codes of Zhang TIP, Zupancic TMM and Jung CSVT are all from the authors and implanted to HM16.5. The scheme of Corrêa CSVT is implemented by

<sup>1</sup><https://drive.google.com/file/d/0B82gacRzADm8NS10NmVQcGltc1k/view?usp=sharing>

TABLE VI  
PERFORMANCE OF FULL FEATURES (LDP) [UNIT:%/DB/%]

| Class | Sequence        | Level 0      |               |             | Level 1      |               |             | Level 2      |               |             | Level 0 + 1 + 2 |               |             |
|-------|-----------------|--------------|---------------|-------------|--------------|---------------|-------------|--------------|---------------|-------------|-----------------|---------------|-------------|
|       |                 | BD BR        | BD PSNR       | TS          | BD BR        | BD PSNR       | TS          | BD BR        | BD PSNR       | TS          | BD BR           | BD PSNR       | TS          |
| B     | BasketballDrive | 0.709        | -0.009        | 30.7        | 1.385        | -0.022        | 38.9        | 1.236        | -0.020        | 25.2        | 2.820           | -0.047        | 51.7        |
|       | BQTerrace       | 0.571        | -0.006        | 35.2        | 0.865        | -0.013        | 37.3        | 0.870        | -0.013        | 21.0        | 2.795           | -0.046        | 50.7        |
|       | Cactus          | 0.486        | -0.007        | 31.6        | 1.214        | -0.020        | 37.6        | 1.038        | -0.020        | 22.7        | 3.028           | -0.054        | 51.9        |
|       | KimonoI         | 0.426        | -0.008        | 20.9        | 1.993        | -0.060        | 48.7        | 0.444        | -0.016        | 28.4        | 2.153           | -0.066        | 54.6        |
| C     | ParkScene*      | 0.334        | -0.006        | 29.0        | 1.041        | -0.024        | 34.9        | 0.867        | -0.022        | 19.6        | 2.298           | -0.057        | 49.3        |
|       | BQMall*         | 0.367        | -0.011        | 20.2        | 1.291        | -0.039        | 30.1        | 1.762        | -0.061        | 21.2        | 3.999           | -0.134        | 48.5        |
|       | PartyScene      | 0.250        | -0.005        | 16.1        | 0.561        | -0.017        | 24.6        | 0.878        | -0.034        | 18.8        | 2.620           | -0.098        | 44.8        |
|       | RaceHorses      | 0.111        | -0.001        | 10.3        | 0.707        | -0.019        | 24.8        | 1.155        | -0.038        | 19.6        | 2.453           | -0.079        | 43.2        |
| D     | BasketballDrill | 0.656        | -0.016        | 25.2        | 1.503        | -0.048        | 36.0        | 1.659        | -0.057        | 23.3        | 4.851           | -0.165        | 52.5        |
|       | BasketballPass* | 0.115        | -0.003        | 15.2        | 0.432        | -0.018        | 28.1        | 2.119        | -0.073        | 23.3        | 2.631           | -0.100        | 47.3        |
|       | BlowingBubbles  | -0.127       | 0.002         | 9.2         | 0.726        | -0.020        | 26.2        | 1.319        | -0.040        | 22.3        | 2.672           | -0.087        | 44.2        |
|       | BQSquare        | 0.192        | -0.013        | 14.4        | 0.316        | -0.011        | 30.4        | 1.051        | -0.040        | 22.9        | 1.574           | -0.044        | 47.0        |
| E     | RaceHorses      | 0.045        | 0.000         | 6.70        | 0.654        | -0.019        | 20.9        | 1.773        | -0.072        | 20.1        | 2.820           | -0.110        | 40.4        |
|       | FourPeople      | 0.388        | -0.007        | 55.0        | 1.530        | -0.037        | 48.4        | 0.907        | -0.028        | 29.1        | 3.039           | -0.077        | 63.8        |
|       | Johnny*         | 0.848        | -0.017        | 55.9        | 2.180        | -0.036        | 50.4        | 1.544        | -0.026        | 28.9        | 3.682           | -0.062        | 64.3        |
|       | KristenAndSara  | 0.811        | -0.019        | 55.5        | 1.461        | -0.033        | 50.1        | 0.722        | -0.023        | 28.9        | 2.628           | -0.062        | 64.0        |
|       | VidyoI          | 0.749        | -0.012        | 56.1        | 1.310        | -0.024        | 49.0        | 1.039        | -0.025        | 28.5        | 2.814           | -0.061        | 63.6        |
|       | <b>AVERAGE</b>  | <b>0.408</b> | <b>-0.008</b> | <b>28.7</b> | <b>1.128</b> | <b>-0.027</b> | <b>36.3</b> | <b>1.199</b> | <b>-0.036</b> | <b>23.8</b> | <b>2.875</b>    | <b>-0.079</b> | <b>51.9</b> |

TABLE VII  
PERFORMANCE OF SELECTED FEATURES (LDP) [UNIT:%/DB/%]

| Class | Sequence        | Level 0      |               |             | Level 1      |               |             | Level 2      |               |             | Level 0 + 1 + 2 |               |             |
|-------|-----------------|--------------|---------------|-------------|--------------|---------------|-------------|--------------|---------------|-------------|-----------------|---------------|-------------|
|       |                 | BD BR        | BD PSNR       | TS          | BD BR        | BD PSNR       | TS          | BD BR        | BD PSNR       | TS          | BD BR           | BD PSNR       | TS          |
| B     | BasketballDrive | 0.562        | -0.007        | 29.3        | 1.237        | -0.020        | 40.1        | 1.179        | -0.019        | 25.2        | 1.718           | -0.028        | 49.0        |
|       | BQTerrace       | 0.327        | -0.004        | 34.4        | 0.533        | -0.008        | 38.1        | 0.594        | -0.011        | 21.9        | 2.052           | -0.033        | 52.3        |
|       | Cactus          | 0.441        | -0.004        | 31.7        | 1.300        | -0.022        | 38.6        | 1.069        | -0.017        | 22.1        | 1.934           | -0.035        | 50.7        |
|       | KimonoI         | 0.419        | -0.008        | 20.9        | 2.652        | -0.084        | 53.9        | 0.407        | -0.015        | 28.1        | 1.180           | -0.039        | 50.2        |
| C     | ParkScene*      | 0.413        | -0.006        | 29.9        | 0.743        | -0.018        | 34.6        | 0.723        | -0.021        | 20.1        | 1.403           | -0.033        | 47.8        |
|       | BQMall*         | 0.196        | -0.004        | 21.1        | 1.153        | -0.034        | 31.7        | 1.943        | -0.061        | 21.4        | 3.088           | -0.102        | 50.4        |
|       | PartyScene      | 0.084        | -0.002        | 11.9        | 0.521        | -0.012        | 26.3        | 0.679        | -0.023        | 18.1        | 2.108           | -0.072        | 48.7        |
|       | RaceHorses      | 0.119        | -0.004        | 12.2        | 0.580        | -0.013        | 26.7        | 1.137        | -0.035        | 19.0        | 1.792           | -0.059        | 44.0        |
| D     | BasketballDrill | 1.326        | -0.026        | 27.1        | 1.202        | -0.039        | 37.7        | 1.339        | -0.044        | 23.3        | 3.817           | -0.123        | 55.5        |
|       | BasketballPass* | 0.272        | -0.009        | 18.6        | 0.405        | -0.012        | 30.0        | 1.533        | -0.066        | 22.4        | 2.336           | -0.091        | 51.3        |
|       | BlowingBubbles  | -0.077       | 0.006         | 14.3        | 0.404        | -0.008        | 28.2        | 1.560        | -0.052        | 23.3        | 2.080           | -0.064        | 47.5        |
|       | BQSquare        | 0.216        | -0.008        | 15.2        | 0.592        | -0.020        | 30.8        | 0.982        | -0.034        | 23.2        | 1.771           | -0.053        | 50.1        |
| E     | RaceHorses      | 0.050        | 0.000         | 7.5         | 0.478        | -0.012        | 22.6        | 1.753        | -0.067        | 18.8        | 2.303           | -0.088        | 42.5        |
|       | FourPeople      | 0.267        | -0.004        | 54.0        | 1.172        | -0.029        | 49.7        | 0.928        | -0.028        | 28.5        | 2.327           | -0.058        | 67.8        |
|       | Johnny*         | 1.294        | -0.005        | 54.6        | 1.426        | -0.028        | 49.9        | 0.642        | -0.018        | 28.4        | 2.673           | -0.046        | 66.4        |
|       | KristenAndSara  | 0.461        | -0.015        | 53.9        | 0.660        | -0.018        | 50.9        | 0.658        | -0.021        | 27.9        | 1.941           | -0.048        | 67.1        |
|       | VidyoI          | 0.499        | -0.009        | 55.0        | 1.102        | -0.022        | 50.4        | 0.682        | -0.018        | 28.2        | 2.079           | -0.044        | 66.9        |
|       | <b>AVERAGE</b>  | <b>0.404</b> | <b>-0.006</b> | <b>28.9</b> | <b>0.950</b> | <b>-0.023</b> | <b>37.7</b> | <b>1.047</b> | <b>-0.032</b> | <b>23.5</b> | <b>2.153</b>    | <b>-0.059</b> | <b>53.4</b> |

TABLE VIII  
PERFORMANCE COMPARISON WITH FOUR STATE-OF-THE-ART FAST ALGORITHMS (LDP) [UNIT:%/DB/%]

| Class | Sequence                  | Correa CSVT [22] |               |             | Zhang TIP [23] |               |             | Zupancic TMM [16] |               |             | Jung CSVT [17] |               |             | Proposed     |               |             |
|-------|---------------------------|------------------|---------------|-------------|----------------|---------------|-------------|-------------------|---------------|-------------|----------------|---------------|-------------|--------------|---------------|-------------|
|       |                           | BD BR            | BD PSNR       | TS          | BD BR          | BD PSNR       | TS          | BD BR             | BD PSNR       | TS          | BD BR          | BD PSNR       | TS          | BD BR        | BD PSNR       | TS          |
| B     | BasketballDrive           | 6.394            | -0.114        | 52.7        | 2.592          | -0.040        | 46.3        | 5.858             | -0.095        | 43.5        | 1.101          | -0.032        | 33.1        | 3.454        | -0.059        | 59.4        |
|       | BQTerrace                 | 1.373            | -0.034        | 48.0        | 1.167          | -0.017        | 45.6        | 3.337             | -0.051        | 46.5        | 0.904          | -0.035        | 33.0        | 2.202        | -0.035        | 58.7        |
|       | Cactus                    | 4.032            | -0.073        | 47.7        | 1.621          | -0.024        | 46.9        | 4.850             | -0.067        | 46.6        | 1.521          | -0.036        | 30.5        | 3.219        | -0.053        | 59.3        |
|       | KimonoI                   | 3.533            | -0.102        | 53.6        | 2.138          | -0.056        | 48.1        | 4.182             | -0.124        | 40.6        | 1.305          | -0.046        | 34.0        | 2.578        | -0.074        | 60.5        |
| C     | ParkScene*                | 3.070            | -0.104        | 46.1        | 1.423          | -0.034        | 45.5        | 3.710             | -0.099        | 46.4        | 0.554          | -0.024        | 29.0        | 2.705        | -0.067        | 56.5        |
|       | BQMall*                   | 8.108            | -0.321        | 39.7        | 1.789          | -0.055        | 40.4        | 4.901             | -0.158        | 48.1        | 0.744          | -0.034        | 17.2        | 3.736        | -0.117        | 55.7        |
|       | PartyScene                | 3.134            | -0.141        | 26.9        | 0.998          | -0.023        | 33.3        | 2.390             | -0.081        | 53.3        | 2.336          | -0.118        | 20.3        | 2.199        | -0.076        | 52.5        |
|       | RaceHorses                | 5.384            | -0.224        | 30.4        | 1.517          | -0.051        | 37.1        | 4.272             | -0.132        | 52.7        | 1.411          | -0.071        | 23.5        | 2.501        | -0.074        | 51.1        |
| D     | BasketballDrill           | 6.610            | -0.266        | 45.6        | 1.551          | -0.048        | 45.4        | 5.171             | -0.180        | 44.2        | 0.660          | -0.032        | 28.0        | 3.345        | -0.111        | 58.9        |
|       | BasketballPass*           | 5.070            | -0.255        | 42.6        | 1.111          | -0.047        | 36.8        | 3.955             | -0.170        | 43.3        | 0.605          | -0.035        | 21.3        | 2.929        | -0.112        | 55.9        |
|       | BlowingBubbles            | 2.693            | -0.111        | 30.3        | 0.736          | -0.022        | 28.8        | 2.942             | -0.094        | 46.3        | 1.346          | -0.066        | 19.4        | 2.494        | -0.083        | 51.4        |
|       | BQSquare                  | 1.719            | -0.080        | 32.3        | 0.929          | -0.026        | 29.6        | 2.812             | -0.093        | 49.9        | 1.303          | -0.059        | 27.0        | 1.694        | -0.052        | 53.3        |
| E     | RaceHorses                | 6.509            | -0.351        | 25.1        | 0.858          | -0.031        | 28.4        | 3.831             | -0.141        | 52.4        | 2.159          | -0.115        | 29.2        | 2.709        | -0.100        | 47.9        |
|       | FourPeople                | 3.490            | -0.121        | 72.9        | 1.866          | -0.048        | 67.3        | 3.415             | -0.086        | 38.2        | -0.354         | 0.004         | 44.6        | 2.806        | -0.067        | 70.3        |
|       | Johnny*                   | 3.226            | -0.079        | 75.2        | 2.485          | -0.044        | 68.6        | 4.242             | -0.070        | 36.4        | -0.490         | -0.003        | 43.0        | 3.914        | -0.062        | 69.8        |
|       | KristenAndSara            | 2.741            | -0.101        | 74.8        | 1.861          | -0.044        | 67.9        | 3.830             | -0.093        | 35.8        | -0.251         | -0.003        | 42.5        | 2.964        | -0.069        | 69.6        |
|       | VidyoI                    | 3.035            | -0.109        | 75.5        | 2.085          | -0.046        | 69.4        | 3.406             | -0.079        | 37.3        | -0.556         | 0.010         | 45.0        | 3.168        | -0.067        | 70.0        |
|       | <b>AVERAGE</b>            | <b>4.125</b>     | <b>-0.152</b> | <b>48.2</b> | <b>1.572</b>   | <b>-0.039</b> | <b>46.2</b> | <b>3.947</b>      | <b>-0.107</b> | <b>44.8</b> | <b>0.841</b>   | <b>-0.041</b> | <b>30.6</b> | <b>2.859</b> | <b>-0.075</b> | <b>58.9</b> |
|       | <b>STANDARD DEVIATION</b> | <b>1.840</b>     | <b>0.091</b>  | <b>16.9</b> | <b>0.544</b>   | <b>0.012</b>  | <b>13.8</b> | <b>0.870</b>      | <b>0.037</b>  | <b>5.49</b> | <b>0.840</b>   | <b>0.035</b>  | <b>8.73</b> | <b>0.559</b> | <b>0.022</b>  | <b>6.96</b> |

ourselves on the platform of HM 16.5. Corr ea CSVT predicts the CTU partitions with decision tree algorithm. Zhang TIP uses two SVMs to predict the CU size in every level, and the different prediction results from two SVMs will activate the

original HEVC test model. Zupancic TMM uses the reverse CU visiting order to reduce the computational complexity. Jung CSVT adopts an adaptive ordering of modes to do computational complexity reduction. Here, the proposed approach

TABLE IX  
PERFORMANCE COMPARISON WITH FOUR STATE-OF-THE-ART FAST ALGORITHMS (RA) [UNIT:%/DB/%]

| Class                     | Sequence        | Correa CSVT [22] |               |             | Zhang TIP [23] |               |             | Zupancic TMM [16] |               |             | Jung CSVT [17] |               |             | Proposed     |               |             |
|---------------------------|-----------------|------------------|---------------|-------------|----------------|---------------|-------------|-------------------|---------------|-------------|----------------|---------------|-------------|--------------|---------------|-------------|
|                           |                 | BD<br>BR         | BD<br>PSNR    | TS          | BD<br>BR       | BD<br>PSNR    | TS          | BD<br>BR          | BD<br>PSNR    | TS          | BD<br>BR       | BD<br>PSNR    | TS          | BD<br>BR     | BD<br>PSNR    | TS          |
| A                         | PeopleOnStreet  | 11.382           | -0.486        | 43.3        | 1.760          | -0.063        | 47.4        | 4.454             | -0.161        | 54.8        | -0.045         | -0.018        | 20.7        | 2.843        | -0.106        | 47.2        |
|                           | Traffic         | 5.035            | -0.170        | 66.5        | 1.937          | -0.050        | 61.6        | 3.125             | -0.095        | 38.3        | 0.053          | -0.006        | 33.0        | 1.997        | -0.060        | 60.3        |
| B                         | BasketballDrive | 7.409            | -0.128        | 62.1        | 4.843          | -0.075        | 51.7        | 4.448             | -0.072        | 38.6        | 0.651          | -0.020        | 27.7        | 3.808        | -0.062        | 57.0        |
|                           | BQTerrace       | 1.629            | -0.041        | 61.5        | 1.380          | -0.022        | 49.9        | 2.153             | -0.038        | 38.7        | 0.358          | -0.013        | 33.0        | 1.540        | -0.029        | 59.8        |
|                           | Cactus          | 5.738            | -0.102        | 60.1        | 2.290          | -0.036        | 51.2        | 3.691             | -0.063        | 40.6        | 0.429          | -0.012        | 24.5        | 3.002        | -0.049        | 56.7        |
|                           | KimonoI         | 4.036            | -0.115        | 65.1        | 4.925          | -0.133        | 52.5        | 2.880             | -0.086        | 34.8        | 1.253          | -0.043        | 27.2        | 2.364        | -0.068        | 57.1        |
|                           | ParkScene*      | 4.370            | -0.149        | 61.0        | 1.717          | -0.046        | 53.2        | 2.814             | -0.081        | 39.9        | 0.288          | -0.011        | 26.1        | 2.260        | -0.062        | 58.3        |
| C                         | BQMall*         | 11.141           | -0.443        | 50.5        | 2.641          | -0.091        | 47.7        | 4.560             | -0.168        | 43.5        | -0.025         | -0.007        | 16.3        | 3.845        | -0.133        | 55.8        |
|                           | PartyScene      | 4.817            | -0.237        | 43.9        | 1.095          | -0.042        | 40.7        | 1.806             | -0.076        | 47.0        | 0.908          | -0.051        | 18.7        | 1.926        | -0.076        | 54.0        |
|                           | RaceHorses      | 7.912            | -0.302        | 38.0        | 3.208          | -0.108        | 41.3        | 4.367             | -0.144        | 50.6        | 1.010          | -0.045        | 20.6        | 3.025        | -0.091        | 51.4        |
|                           | BasketballDrill | 7.979            | -0.343        | 54.8        | 1.919          | -0.058        | 50.7        | 3.728             | -0.148        | 39.6        | 0.076          | -0.008        | 21.8        | 3.116        | -0.107        | 57.6        |
|                           | BasketballPass* | 8.439            | -0.426        | 53.2        | 2.734          | -0.130        | 40.5        | 3.025             | -0.137        | 40.2        | 0.103          | -0.009        | 21.4        | 3.275        | -0.141        | 55.8        |
| D                         | BlowingBubbles  | 4.492            | -0.196        | 44.8        | 0.991          | -0.030        | 33.7        | 2.487             | -0.093        | 41.6        | 0.611          | -0.030        | 15.7        | 2.511        | -0.091        | 52.4        |
|                           | BQSquare        | 2.742            | -0.142        | 54.3        | 0.644          | -0.027        | 35.3        | 1.253             | -0.052        | 40.0        | 0.062          | -0.007        | 23.3        | 1.263        | -0.059        | 58.3        |
|                           | RaceHorses      | 10.689           | -0.538        | 33.2        | 2.177          | -0.101        | 33.1        | 3.811             | -0.157        | 50.9        | 1.425          | -0.082        | 25.0        | 3.292        | -0.142        | 48.3        |
|                           | <b>AVERAGE</b>  | <b>6.521</b>     | <b>-0.255</b> | <b>52.8</b> | <b>2.284</b>   | <b>-0.068</b> | <b>46.0</b> | <b>3.240</b>      | <b>-0.105</b> | <b>42.6</b> | <b>0.477</b>   | <b>-0.024</b> | <b>23.7</b> | <b>2.671</b> | <b>-0.085</b> | <b>55.3</b> |
| <b>STANDARD DEVIATION</b> |                 | <b>2.941</b>     | <b>0.152</b>  | <b>9.91</b> | <b>1.215</b>   | <b>0.036</b>  | <b>7.96</b> | <b>0.995</b>      | <b>0.042</b>  | <b>5.45</b> | <b>0.465</b>   | <b>0.021</b>  | <b>5.04</b> | <b>0.750</b> | <b>0.033</b>  | <b>3.80</b> |

is equipped with the classifiers trained by Eq.(18) and the width of risk area is determined by Eqs.(19) and (27). It should be noted that the classifiers are only trained with the training data from the first GOP of every sequence without updating. The experimental results are shown in Tables VIII and IX under LP and RA configurations, respectively.

From Table VIII, it can be found that, 48.2% average computational complexity is reduced with the RD performance loss as 4.125% in terms of BD-BR and -0.152dB in terms of BD-PSNR for Corrêa CSVT, Zhang TIP can reduce 46.2% computational complexity on average with the values of BD-BR and BD-PSNR as 1.572% and -0.039dB, Zupancic TMM achieves 44.8% TS while the bit rate increases 3.947% and the PSNR loss reaches -0.107dB on average, Jung CSVT keeps the great RD performance but obtains the least computational complexity reduction, while for the proposed approach, it achieves 58.9% computational complexity reduction and keeps the values of BD-BR and BD-PSNR as 2.859% and -0.075dB. Compared with Zhang TIP, the proposed approach achieves 12.7% complexity reduction and 1.287% BD-rate increase on average under LDP configuration. The reasons are that the BD-rate does not increase in a linear manner with the complexity reduction, and it becomes more and more difficulty to reduce the complexity and maintain the better RD performance after the complexity has been largely explored. In fact, the proposed approach has less classifiers utilization, more flexible performance control and more factors consideration (outliers and misclassification cost). Because the performance of proposed approach can be easily adjusted, *i.e.*, aiming at maximizing the complexity reduction under a configurable RD performance degradation constraint. We change the RD performance degradation constraint, which is set as similar to the RD performance of Zhang TIP. The experimental results reveal that the proposed method is a little better than Zhang TIP in terms of complexity reduction and RD performance, the BD-BR of the former is 1.411% and the BD-BR of the latter is 1.572%, while the time saving of the former is 49.7% and the time saving of the latter is 46.2%. At the same time, the values of standard deviation for each scheme are calculated in terms of RD performance loss and TS, which demonstrate that Zhang

TIP and the proposed approach have the stable RD performance, while Zupancic TMM and the proposed approach keep a little computational complexity reduction fluctuation. Similar results can be found under RA configuration in Table IX.

Among these state-of-the-art algorithms, Corrêa CSVT just takes decision tree algorithm for early termination and CU skipping, the RDC optimization is not effectively considered. Zhang TIP and Jung CSVT both have a limited TS because they use a strict constraint for the RD performance loss. For the scheme of Zupancic TMM, the settings of ESD, ECU, and CFM are inactivated for a fair performance comparison, which means that it has a limited contribution on computational complexity reduction by their own scheme. Compared with these algorithms, the proposed approach makes the best RDC performance.

In addition, the proposed method is compared with the HEVC test model which is equipped with enabled ESD, ECU and CFM. The results show that the proposed method can further reduce 9.7% complexity with only about 0.9% bit rate increase under RA configuration. Also, the proposed method has more stable complexity reduction for different sequences.

### C. Percentage of Time Consumption in Proposed Approach

To test the complexity overhead of classifiers training in the proposed approach, the percentages of time consumption are recorded. Four sequences with different resolutions and different contents, *i.e.*, BlowingBubbles (416×240), PartyScene (832×480), FourPeople (1280×720), and BasketballDrive (1920×1080), are adopted and encoded under LDP configuration at QP 27. The results are shown in Fig. 7. In the proposed approach, two important parts are included, namely, classifiers training and video encoding. According to Fig. 7, it can be found that the classifiers training only occupies about 1% in total time, which can be ignored. Therefore, the complexity overhead of classifiers training has no impact on the complexity reduction. In addition, for the sequence with larger resolution, the time consumption of classifiers training is more

TABLE X  
PERFORMANCE COMPARISON IN CASE OF SCENE CHANGE (LDP) [UNIT:%/dB/%]

| Sequence       | Correa CSVT [22] |               |             | Zhang TIP [23] |               |             | Zupancic TMM [16] |               |             | Jung CSVT [17] |               |             | Proposed     |               |             |
|----------------|------------------|---------------|-------------|----------------|---------------|-------------|-------------------|---------------|-------------|----------------|---------------|-------------|--------------|---------------|-------------|
|                | BD<br>BR         | BD<br>PSNR    | TS          | BD<br>BR       | BD<br>PSNR    | TS          | BD<br>BR          | BD<br>PSNR    | TS          | BD<br>BR       | BD<br>PSNR    | TS          | BD<br>BR     | BD<br>PSNR    | TS          |
| Seq1           | 4.702            | -0.211        | 28.8        | 0.984          | -0.027        | 29.0        | 3.278             | -0.109        | 49.4        | 1.126          | -0.060        | 12.8        | 2.843        | -0.096        | 51.5        |
| Seq2           | 5.747            | -0.233        | 36.9        | 1.566          | -0.043        | 40.9        | 3.827             | -0.126        | 48.7        | 1.640          | -0.080        | 19.6        | 3.238        | -0.102        | 55.9        |
| Seq3           | 5.276            | -0.164        | 73.4        | 3.024          | -0.070        | 64.8        | 4.054             | -0.099        | 37.0        | -0.434         | 0.004         | 39.2        | 5.127        | -0.115        | 70.3        |
| Seq4           | 4.695            | -0.077        | 48.0        | 2.063          | -0.027        | 46.3        | 4.766             | -0.067        | 44.5        | 0.687          | -0.023        | 23.7        | 3.232        | -0.047        | 57.8        |
| <b>AVERAGE</b> | <b>5.105</b>     | <b>-0.171</b> | <b>46.8</b> | <b>1.909</b>   | <b>-0.042</b> | <b>45.2</b> | <b>3.982</b>      | <b>-0.100</b> | <b>44.9</b> | <b>0.755</b>   | <b>-0.040</b> | <b>23.8</b> | <b>3.609</b> | <b>-0.089</b> | <b>58.9</b> |

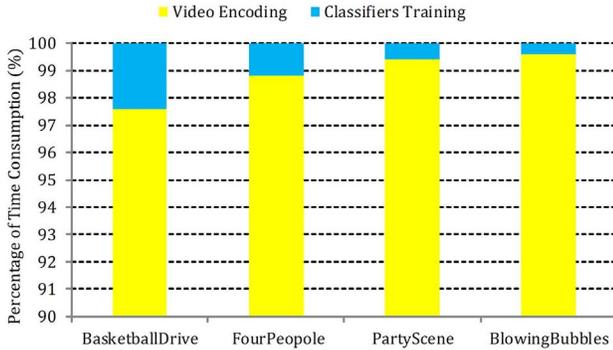


Fig. 7. Percentage of time consumption in the proposed approach.

than that of others, because it has more training samples when the number of training frames is fixed.

#### D. Performance Verification in Case of Drastic Scene Change

Additionally, four new sequences with drastic scene change are generated for performance verification. Three sequences with the same resolution from traditional sequences are combined together for each new sequence. For example, Seq1 (416×240) is created by combining BlowingBubbles, RaceHorses and BQSquare in order. In detail, the first 101 frames are extracted from BlowingBubbles, the following 100 frames are extracted from RaceHorses, and the last 100 frames are extracted from BQSquare. There are 301 frames (0-300) in total and the scene change will occur at the 101<sup>th</sup> and 201<sup>th</sup> frames. Similarly, Seq2 (832×480) is created from PartyScene, BasketballDrill and BQMall, Seq3 (1280×720) is created from FourPeople, KristenAndSara and Vidy01, and Seq4 (1920×1080) is created from BasketballDrill, BQTerrace and Cactus. These four new sequences are encoded under the same configuration as the traditional sequences.

Here, the Classifiers #0, #1 and #2 are trained with the training data from the first GOP of current sequence and periodically updated (120 frames per period in this experiment) for the proposed method. The periodically updating means that Classifiers #0, #1 and #2 will be re-trained at the 121<sup>th</sup> frame, the 241<sup>th</sup> frame, the 361<sup>th</sup> frame and so on. It should be noted that the classifiers are not updated after scene change, but periodically updated, because we do not know when the scene change will happen in practice. The experimental results are illustrated in Table X.

In case of drastic scene change, Corrêa CSVT saves about 46.8% time and the RD performance degradation reaches

5.105% in terms of BD-BR and -0.171dB in terms of BD-PSNR on average, Zhang TIP can reduce 45.2% computational complexity with the values of BD-BR and BD-PSNR as 1.909% and -0.042dB, Zupancic TMM reduces about 44.9% time with the BD-BR and BD-PSNR as 3.982% and -0.100dB, Jung CSVT achieves about 23.8% computational complexity reduction with the values of BD-BR and BD-PSNR as 0.755% and -0.040dB. While for the proposed approach, it can reduce about 58.9% computational complexity with the RD performance degradation as 3.609% in terms of BD-BR and -0.089dB in terms of BD-PSNR. From the experimental results, it can be easily found that the proposed method outperforms the state-of-the-art algorithms as same as the traditional sequences.

## VII. CONCLUSION

This paper presents a CU decision method based on ML, in which the quadtree structure of CTU partition is modeled as a multi-level cascaded classification task and the CUs can be directly predicted by a fuzzy SVM. Different from traditional classification problems, the proposed CU decision scheme incorporates a misclassification cost defined by the combination of classification prediction accuracy and RD performance degradation. Then, with the misclassification cost, the optimal feature set is selected, and the regulation parameters of SVM are determined. At the same time, the negative influence from outlier in the training data is eliminated. Additionally, in order to achieve a great RDC performance, a risk area is introduced into the SVM decision function, which means that if the current CU belongs to the risk area, it will activate the original HEVC test model. Besides, the different widths of risk area are allocated to different CUs aiming at maximizing the computational complexity reduction under a configurable RD performance degradation constraint. Compared with the state-of-the-art statistical information and ML based fast algorithms, the proposed method is the best in terms of RDC performance.

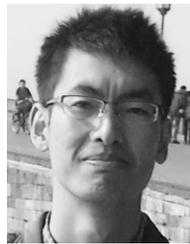
## ACKNOWLEDGMENT

The authors would like to thank Dr. Ivan Zupancic and Dr. Soon-Heung Jung for sharing their source codes and thank JCT-VC for its high quality test sequences.

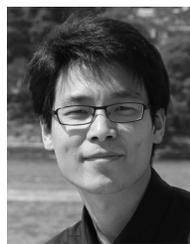
## REFERENCES

- [1] G. J. Sullivan, J. 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] T. Wiegand, G. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

- [3] G. Corrêa, P. A. Assunção, L. V. Agostini, and L. A. da Silva Cruz, "Pareto-based method for high efficiency video coding with limited encoding time," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1734–1745, Sep. 2016.
- [4] X. Deng, M. Xu, L. Jiang, X. Sun, and Z. Wang, "Subjective-driven complexity control approach for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 91–106, Jan. 2016.
- [5] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1697–1706, Dec. 2012.
- [6] Z. Pan, S. Kwong, M.-T. Sun, and J. Lei, "Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 405–412, Jun. 2014.
- [7] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast motion estimation based on content property for low-complexity H.265/HEVC encoder," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 675–684, Sep. 2016.
- [8] Z. Pan *et al.*, "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.
- [9] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machine," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.
- [10] 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.
- [11] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 465–470, Feb. 2013.
- [12] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 422–435, Mar. 2015.
- [13] J. Xiong, H. Li, F. Meng, Q. Wu, and K. N. Ngan, "Fast HEVC inter CU decision based on latent SAD estimation," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2147–2159, Dec. 2015.
- [14] B. Lee, J. Jung, and M. Kim, "An all-zero block detection scheme for low-complexity HEVC encoders," *IEEE Trans. Multimedia*, vol. 18, no. 7, pp. 1257–1268, Jul. 2016.
- [15] H. Fan *et al.*, "Hybrid zero block detection for high efficiency video coding," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 537–543, Mar. 2016.
- [16] 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.
- [17] 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.
- [18] H. Lee, H. J. Shim, Y. Park, and B. Jeon, "Early skip mode decision for HEVC encoder with emphasis on coding quality," *IEEE Trans. Broadcast.*, vol. 61, no. 3, pp. 388–397, Sep. 2015.
- [19] Y. Li, G. Yang, Y. Zhu, X. Ding, and X. Sun, "Unimodal stopping model-based early SKIP mode decision for high-efficiency video coding," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1431–1441, Jul. 2017.
- [20] J. Zhang, B. Li, and H. Li, "An efficient fast mode decision method for inter prediction in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 8, pp. 1502–1515, Aug. 2016.
- [21] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP J. Image Video Process.*, vol. 2013, pp. 1–11, Jan. 2013.
- [22] G. Corrêa, P. A. Assunção, 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.
- [23] Y. Zhang *et al.*, "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.
- [24] H.-S. Kim and R.-H. Park, "Fast CU partitioning algorithm for HEVC using an online-learning-based Bayesian decision rule," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 130–138, Jan. 2016.
- [25] Q. Hu, X. Zhang, Z. Shi, and Z. Gao, "Neyman–Pearson-based early mode decision for HEVC encoding," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 379–391, Mar. 2016.
- [26] L. Zhu *et al.*, "Binary and multi-class learning based low complexity optimization for HEVC encoding," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 547–561, Sep. 2017.
- [27] T. Zhang, M.-T. Sun, D. Zhao, and W. Gao, "Fast intra-mode and CU size decision for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1714–1726, Aug. 2017.
- [28] F. Duanmu, Z. Ma, and Y. Wang, "Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension," *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp. 517–531, Dec. 2016.
- [29] Y. Zhang *et al.*, "Effective data driven coding unit size decision approaches for HEVC INTRA coding," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: [10.1109/TCSVT.2017.2747659](https://doi.org/10.1109/TCSVT.2017.2747659).
- [30] Z. Liu *et al.*, "CU partition mode decision for HEVC hardwired intra encoder using convolution neural network," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5088–5103, Nov. 2016.
- [31] L. Zhu, Y. Zhang, N. Li, G. Jiang, and S. Kwong, "Machine learning based fast H.264/AVC to HEVC transcoding exploiting block partition similarity," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 824–837, Jul. 2016.
- [32] D. Moraes, J. Wainer, and A. Rocha, "Low false positive learning with support vector machines," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 340–350, Jul. 2016.
- [33] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 464–471, Mar. 2002.
- [34] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document M33, ITU-T Video Coding Experts Group, Austin, TX, USA, 2001.
- [35] C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. J. Sullivan, *High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Update 2 of Encoder Description*, document JCTVC-T1002, JCT-VC ITU-T SG16 WP3, Geneva, Switzerland, Feb. 2015. [Online]. Available: [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/HM-16.5/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.5/)
- [36] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-J1100, JCTVC ITU-T SG16 WP3, Stockholm, Sweden, Jul. 2012.



mainly include depth image-based rendering, depth estimation, and video coding/transcoding.



of Advanced Technology, CAS, where he has been serving as an Associate Professor since 2012. His research interests are video compression, 3-D video processing, and visual perception.



Department of Computer Science. His research interests are video and image coding and evolutionary algorithms.

**Linwei Zhu** (S'16) received the B.S. degree in

applied physics from the Tianjin University of Technology, China, in 2010, and the M.S. degree in signal and information processing from Ningbo University, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. From 2011 to 2013, he was a visiting student with the Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Science. He joined the SIAT as a Research Assistant. His research interests

**Yun Zhang** (M'12–SM'16) received the B.S.

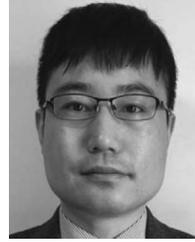
and M.S. degrees in electrical engineering from Ningbo University, Ningbo, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2009 to 2014, he was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong, Hong Kong. In 2010, he became an Assistant Professor with the Shenzhen Institutes

**Sam Kwong** (F'13) received the B.S. degree in

electrical engineering from the State University of New York at Buffalo in 1983, the M.S. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1985, and the Ph.D. degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada. He joined Bell Northern Research Canada as a Scientific Staff Member. In 1990, he became a Lecturer with the Department of Electronic Engineering, City



**Xu Wang** (M'15) received the B.S. degree from South China Normal University, Guangzhou, China, in 2007, the M.S. degree from Ningbo University, Ningbo, China, in 2010, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014. In 2015, he joined the College of Computer Science and Software Engineering, Shenzhen University as an Assistant Professor. His research interests are video coding and stereoscopic image/video quality assessment.



**Tiesong Zhao** (S'08–M'12) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2006 and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2011. He has served as a Research Associate with the Department of Computer Science, City University of Hong Kong, from 2011 to 2012, a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, from 2012 to 2013, and a Research Scientist with the Ubiquitous Multimedia Laboratory, State University of New York at Buffalo until 2015. He is currently a Professor with the College of Physics and Information Engineering, Fuzhou University, Fujian, China. His research interests include image/video signal processing, visual quality assessment, and video coding and transmission.