Online Learning-Based Multi-Stage Complexity Control for Live Video Coding

Chao Huang, Zongju Peng, *Member, IEEE*, Yong Xu, *Senior Member, IEEE*, Fen Chen, Qiuping Jiang, *Member, IEEE*, Yun Zhang, *Senior Member, IEEE*, Gangyi Jiang, *Senior Member, IEEE*, and Yo-Sung Ho, *Fellow, IEEE*

***Abstract*— High Efficiency Video Coding (HEVC) can significantly improve the compression efficiency in comparison with the preceding H.264/Advanced Video Coding (AVC) but at the cost of extremely high computational complexity. Hence, it is challenging to realize live video applications on low-delay and power-constrained devices, such as the smart mobile devices. In this article, we propose an online learning-based multi-stage complexity control method for live video coding. The proposed method consists of three stages: multi-accuracy Coding Unit (CU) decision, multi-stage complexity allocation, and Coding Tree Unit (CTU) level complexity control. Consequently, the encoding complexity can be accurately controlled to correspond with the computing capability of the video-capable device by replacing the traditional brute-force search with the proposed algorithm, which properly determines the optimal CU size. Specifically, the multi-accuracy CU decision model is obtained by an online learning approach to accommodate the different characteristics of input videos. In addition, multi-stage complexity allocation is implemented to reasonably allocate the complexity budgets to each coding level. In order to achieve a good trade-off between complexity control and rate distortion (RD) performance, the CTU-level complexity control is proposed to select the optimal accuracy of the CU decision model. The experimental results show that the proposed algorithm can accurately control the coding complexity from 100% to 40%. Furthermore, the proposed algorithm outperforms the state-of-the-art algorithms in terms of both accuracy of complexity control and RD performance.**

***Index Terms*— Complexity allocation, complexity control, high efficiency video coding, online learning, random forest.**

Manuscript received November 12, 2018; revised October 26, 2019; accepted October 18, 2020. Date of publication November 16, 2020; date of current version December 4, 2020. This work was supported in part by the Natural Science Foundation of China under Grant 61771269, Grant 61620106012, Grant 61901236, Grant 61871247, and Grant 61931022; in part by the Establishment of Key Laboratory of Shenzhen Science and Technology Innovation Committee under Grant ZDSYS20190902093015527; in part by the Natural Science Foundation of Zhejiang Province under Grant LY20F010005; in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K202001105; and in part by the Scientific Research Foundation of Chongqing University of Technology. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jana Ehmann. *(Corresponding author: Zongju Peng.)*

Chao Huang is with the Harbin Institute of Technology at Shenzhen, Shenzhen 518055, China, also with the Peng Cheng Laboratory, Shenzhen 518055, China, and also with the Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China (e-mail: huangchao_08@126.com).

Zongju Peng and Fen Chen are with the School of Electrical and Electronic Engineering, Chongqing University of Technology, Chongqing 400054, China, and also with the Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China (e-mail: pengzongju@126.com; chenfen1@cqut.edu.cn).

Yong Xu is with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen 518055, China, and also with the Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: yongxu@ymail.com).

Qiuping Jiang and Gangyi Jiang are with the Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China (e-mail: jiangqiuping@nbu.edu.cn; jianggangyi@126.com).

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

Yo-Sung Ho is with the Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, South Korea (e-mail: hoyosung@gmail.com).

Digital Object Identifier 10.1109/TIP.2020.3036766

I. Introduction

OVER the last decade, the demand for live video services has seen explosive growth, especially after the popularity of video-capable mobile devices (smartphones and laptops, etc.). Live video applications have been widely utilized in various fields such as remote education, online games, video conferencing, and online video chats, owing to their ability to provide real-time user experience. Meanwhile, the resolution of live videos has been continuously increasing because High Definition (HD) and Ultra-HD (UHD) video services can provide users with high quality immersive experience and more realistic visual enjoyment. However, the amount of data enormously increases which introduces new challenge to live video transmission. Therefore, the Joint Collaborative Team on Video Coding (JCT-VC) developed the High Efficiency Video Coding (HEVC) standard [1]. Compared with the preceding H.264/Advanced Video Coding (AVC) standard [2], HEVC can achieve 50% transmission bitrate reduction while maintaining the same subjective visual quality [3]–[5]. Unfortunately, the encoding performance gain is at the cost of intensive computational complexity. Compared with H.264/AVC, the computational complexity of HEVC has increased by 253% on average [3].

So far, many researchers have focused on low complexity encoding optimization [6]–[21]. In general, these low complexity encoding algorithms aim to save as much encoding time as possible with negligible Rate Distortion (RD) performance degradation. In principle, these algorithms aim to predict the optimal encoding parameters instead of performing a brute-force RD Optimization (RDO) search. However, low-complexity encoding algorithms have certain limitations:

1) The computational complexity reduction is not necessarily adaptable to video-capable devices with different computing capabilities because these fast algorithms aim to ensure the encoding efficiency.
2) Low-complexity encoding algorithms cannot flexibly and effectively achieve a tunable trade-off between computational complexity and encoding efficiency based

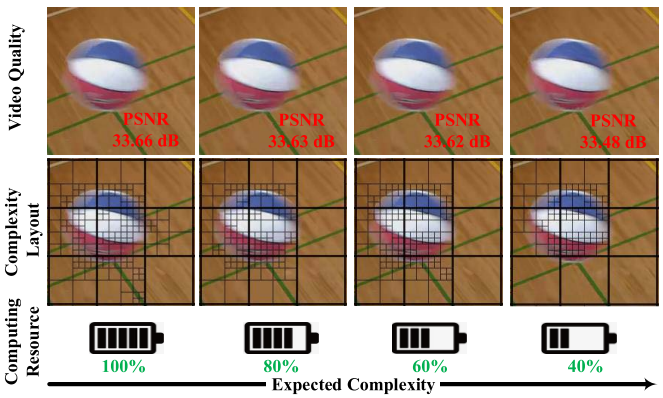


Fig. 1. Illustration of the relation between the visual quality and complexity control. When the devices are with sufficient computing resource, video is encoded with high quality. Once the computing resource is insufficient, video is encoded with low computational complexity (low quality).

on the computing capabilities of different video-capable devices.

3) The computational complexity reduction of the fast encoding algorithm fluctuates greatly depending on the video content. In other words, the fast encoding algorithm cannot guarantee that all video sequences are effectively encoded under computational complexity constraint.

In addition, live video applications are usually performed on mobile devices with different computing capacities and constrained power. Therefore, the tunable trade-off between encoding complexity reduction and compression efficiency is necessary for the encoding algorithm to adapt to different devices. Fig. 1 illustrates the relation between the visual quality and computational consumption in which the percentage is calculated by using the encoding time of original encoding platform as benchmark. Clearly, if more computational power is allocated for encoding, CU splitting is more elaborate. Consequently, the prediction is more accurate, and the video quality improves. Actually, complexity control algorithms have attracted increasing attention from both academia and industry with the aim of addressing these problems. The complexity control algorithm attempts to obtain optimal encoding efficiency for a given expected complexity, which can be adaptively set according to the computing capabilities of different video-capable devices. To achieve this goal, we propose an online learning-based multi-stage complexity control method. The framework of the proposed method is shown in Fig. 2. In this method, a random forest-based multi-accuracy Coding Unit (CU) decision model is first obtained by the online learning approach. Subsequently, multi-stage complexity allocation is designed to assign the complexity budgets to four coding levels, segment level, Group of Pictures (GOP) level, frame level, and Coding Tree Unit (CTU) level. It is noted that the first frame of each segment (update frame) is encoded by the original HEVC encoder and used to online update the parameters of the multi-stage complexity allocation model. Finally, a CTU-level complexity control method is proposed to control the complexity reduction of each CTU by adjusting the accuracy of the CU decision model. The contributions of the proposed method are summarized as follows:

1) To our best knowledge, we are the first to adopt an online learning technique to overcome the complexity control problem in video coding by reasonably adjusting the accuracy of the CU decision model.
2) Differing from existing complexity control methods [22]–[30], the proposed method adopts a multi-stage complexity allocation strategy to achieve a better trade-off between control accuracy and RD performance. This strategy enables the complexity budgets to be more reasonably allocated to each level.
3) The CTU-level complexity control is well designed to make the actual encoding complexity of each CTU adaptive to the expected complexity. A depth-complexity model is proposed to achieve a good trade-off between computational complexity and RD performance.
4) A feedback mechanism is proposed to eliminate the complexity control error caused by inappropriate complexity allocation.

The remainder of this article is organized as follows. Section II introduces motivation and related work. Section III details the random forest-based multi-accuracy CU decision model. Section IV describes the scheme for multi-stage complexity control. Section V presents the experimental results and discussions. Finally, Section VI concludes the paper.

II. Motivation and Related Works

A. Encoding Complexity Analysis

This work aims to achieve complexity control for live video coding based on the HEVC platform, so it is expected to adopt the low-delay P main configuration [31] to achieve low encoding time delay. In the default Low-delay Hierarchical Coding Structure (LDHCS) of HEVC, all frames in a video sequence are divided into a series of GOPs. Note that the first GOP contains only one Intra (I) frame. Furthermore, the number corresponding to each frame indicates the encoding order denoted as Picture Order Count (POC) in the LDHCS. By contrast, each of the subsequent GOPs is composed of four Prediction (P) frames as shown in Fig. 3. The Quantization Parameter (QP) varies with the layer, which indicates that frames in different layers have different computational complexity. Thus, it is necessary to adopt different complexity allocation strategies for frames in different layers.

In HEVC, the Quad-Tree (QT) partition structure is one of the main contributors to compression efficiency gain, with CU partition and multiple prediction modes selection as the core technologies. In the main profile of HEVC, each frame is partitioned into multiple CTUs with the size of 64×64 pixels. A CTU can be further partitioned into multiple CUs via the QT partition structure, as shown in Fig. 4. The optimal coding depths of the CUs in each CTU are determined by recursive RDO search, including a top-down checking process and a bottom-up pruning process. Actually, the recursive CU checking is primarily responsible for the high computational complexity in the brute-force RDO search. For a CTU, it is necessary to check 85 CUs, which include 1, 4, 4^2, and 4^3 CUs with depths of 0, 1, 2, and 3, respectively. However, in the optimal partition combination for a CTU, only certain CUs are selected, from 1 (if the 64×64 CU is not partitioned) to 64 (if the whole CTU is partitioned into 8×8 CUs) [18]. This means that at least 21 CUs and at most 84 CUs do not need to be

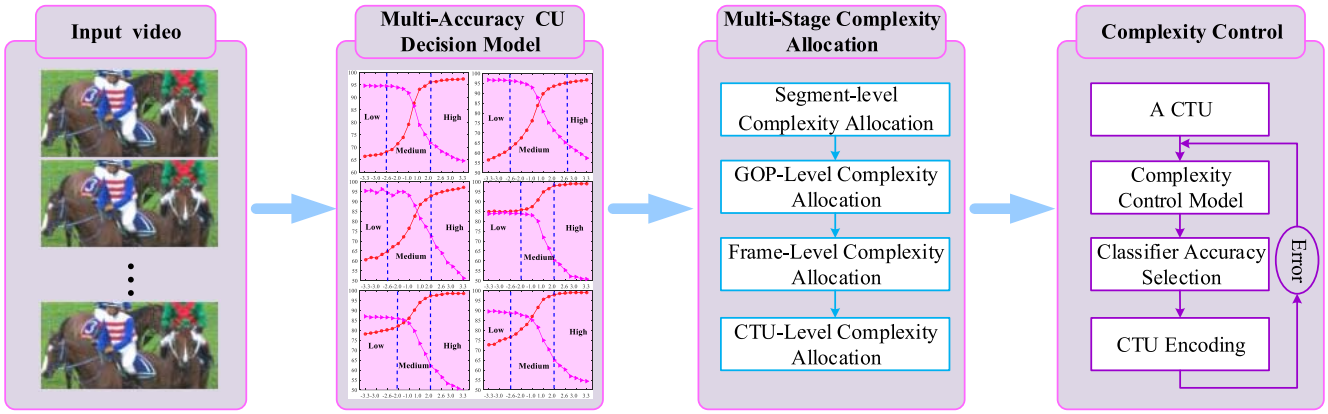


Fig. 2. Overview of online learning-based multi-stage complexity control for live video coding.

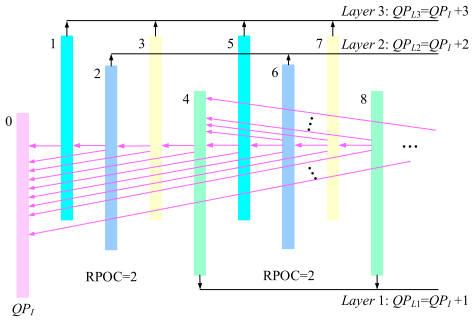


Fig. 3. Hierarchical coding structure under low-delay P configuration [31].

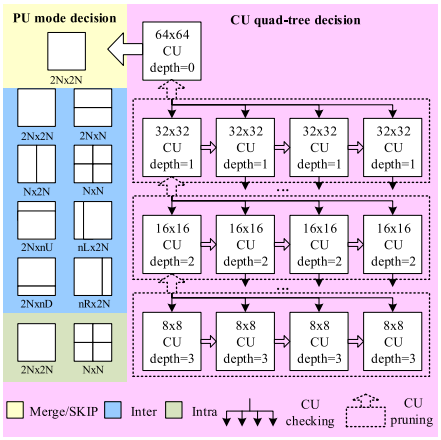


Fig. 4. Quad-tree coding structure in HEVC Inter coding.

checked as a result of the efficient prediction of the CU depth decision. Eliminating redundant RDO search (unnecessary CU checking), achieving expected computational complexity, and maintenance of optimal RD performance are the motivation of this work.

We have statistically analyzed the encoding complexity of HEVC at the GOP layer. Five video sequences with different contents and resolutions were tested on the HM-16.0 under the low-delay P main configuration. Fig. 5 shows the frame-level complexity of tested video sequences. Clearly, the encoding complexity varies with the video resolution and video contents. More importantly, the encoding complexity of these sequences regularly vary, with a period of 4 frames. In other words, the encoder approximately takes the same time to encode each GOP. This phenomena can be used to achieve the GOP-level complexity allocation.

In addition, frames with same Relative POC (RPOC) in different GOPs are encoded with almost the same complexity. To facilitate the analysis of the frame-level encoding complexity, we defined the complexity ratio of the i-th frame in the G-th GOP as

$$\beta_G^{Frame}(i) = T_G^{Frame}(i)/T_G^{GOP} = T_G^{Frame}(i)\Big/\sum_{i=1}^{4} T_G^{Frame}(i), \quad (1)$$

where $T_G^{Frame}(i)$ is the encoding time of the i-th frame in the G-th GOP and $i = 1, 2, 3, 4$, and T_G^{GOP} is the encoding time of G-th GOP.

Fig. 6 shows the complexity ratios of five test sequences. We have two important observations: 1) $\beta_G^{Frame}(i)$ of different sequences with same QP are approximately same; 2) $\beta_G^{Frame}(i)$ of the same sequence weakly varies with the value of QP. Therefore, the frame-level complexity allocation can be transformed into the problem of obtaining $\beta_G^{Frame}(i)$ in a GOP.

B. Related Works

Many efforts have previously been made for encoding complexity optimization. In general, the corresponding researches can be divided into two categories: 1) complexity reduction and 2) complexity control.

With regard to complexity reduction, many researchers aim to optimize the flexible QT partitioning structure, which is mainly responsible for the high computational complexity in HEVC. The basic idea is to predict the optimal CU depth instead of conducting brute-force RDO search [6]–[9]. Specifically, Shen *et al.* [6] proposed an efficient CU depth decision algorithm, in which the texture complexity and high-value coded information of temporal neighboring CUs are used to narrow the CU depth search range. In [7], a prediction mode and CU depth decision scheme are proposed, in which the RD cost of the Skip/Merge mode and parent CU prediction mode are utilized to speed up the process of CU decision. Moreover, some works on fast PU mode decision and Transform Unit (TU) size decision have been proposed to reduce the encoding

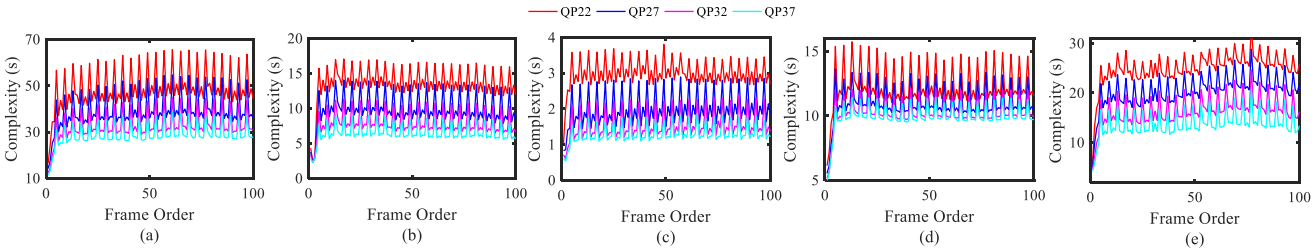


Fig. 5. Frame-level complexity from five test video sequences. (a) *ParkScene*; (b) *PartyScene*; (c) *BQSquare*; (d) *Vidyo1*; (e) *ChinaSpeed*.

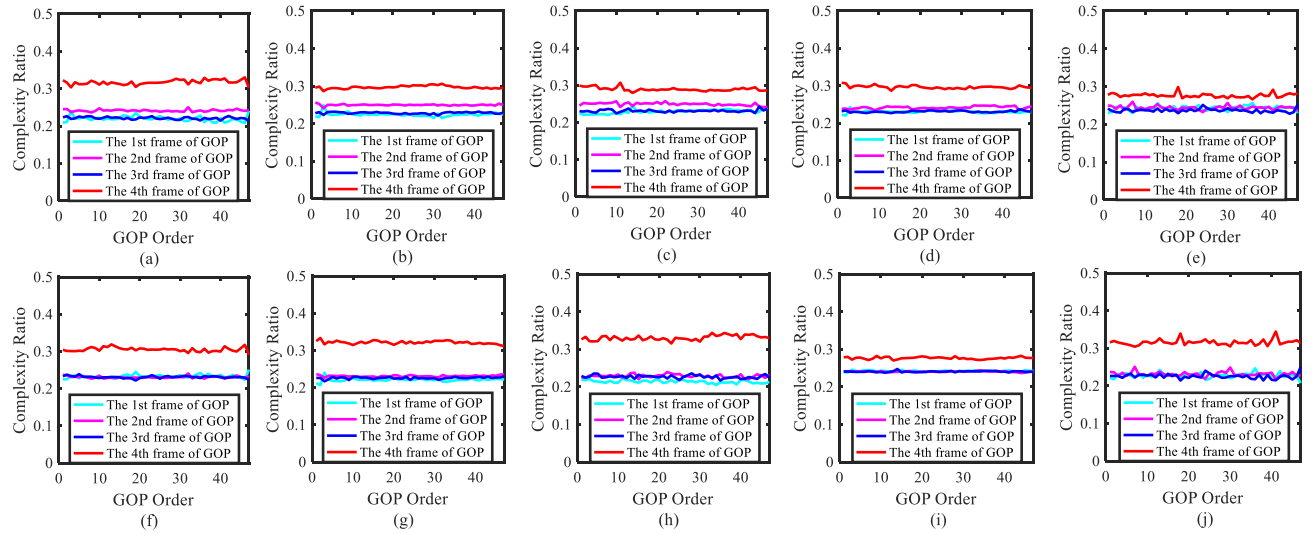


Fig. 6. Frame-level complexity ratio for each GOP. (a) *ParkScene* (QP27); (b) *PartyScene* (QP27); (c) *BQSquare* (QP27); (d) *Vidyo1* (QP27); (e) *ChinaSpeed* (QP27); (f) *ParkScene* (QP37); (g) *PartyScene* (QP37); (h) *BQSquare* (QP37); (i) *Vidyo1* (QP37); (j) *ChinaSpeed* (QP37).

complexity [10]–[13]. For instance, Ryu and Kang [10] utilized random forest to estimate the intra-prediction mode of PU and to achieve encoding complexity reduction by avoiding the complex RDO search of a number of unnecessary modes. In addition, some algorithms aim to accelerate other modules in HEVC, such as fast motion estimation [14], fast reference frame selection [15], and low-complexity in-loop filtering [16]. Most recently, machine learning has become a hotspot and has also been applied in video coding to obtain more encoding complexity reduction [17]–[21]. For example, Xu *et al.* [18] proposed a deep learning-based CU depth decision algorithm to reduce the computational complexity of both intra and inter coding in HEVC, in which convolutional neural network and long- and short-term memory network are applied to predict the optimal CU partition.

Compared with the large number of complexity reduction algorithms, research on encoding complexity control is still at the initial exploratory stage. In general, the main ideas of existing complexity control methods can be divided into three categories: 1) process simplification, 2) encoding parameter adjustment, and 3) complexity allocation. As for encoding process simplification, the basic goal is to achieve the expected complexity by early skipping some encoding parameters decision or early terminating some exhaustive traversal processes. Encoding parameter adjustment aims to establish a parameter-complexity model and control the complexity by adjusting one or more encoding parameters. By contrast, complexity allocation aims to ensure that the actual encoding complexity satisfies the requirement by reasonably allocating the complexity resources. Actually, in order to achieve the expected complexity, the existing works may adopt one or more of these ideas.

Specifically, Corrêa *et al.* [22], [23] utilized the temporal correlation between two frames to predict the maximum CU partition depth. They classify frames into two categories: unconstrained and constrained. The maximum depth in constrained frames is limited by the actual maximum partition depth of the corresponding neighboring CTUs in the nearest unconstrained frame. In this way, the expected complexity can be achieved by adjusting the number of constrained frames. However, the RD performance degradation of sequences with complex texture is significant. Similarly, another proposed algorithm [24] also achieves the expected complexity by constraining the maximum CU partition depth. This method explores the relationship between visual distortion and maximum CU depth to predict the number of CTUs with different depths. Finally, the maximum depths can be allocated to each CTU according to their saliency weights. However, this method only achieves complexity control at the CTU level and the complexity control accuracy is not satisfactory. In order to improve the control accuracy, Deng *et al.* [25] further exploited a hierarchical complexity control method. In this approach, frame-level complexity allocation strategy is proposed, and the coding bit consumption of the previous

frame is utilized to allocate the complexity budgets to the current CTU. Jimenez-Moreno *et al.* [26] followed a different approach by proposing parameter adjustment-based complexity control for HEVC, in which a feedback mechanism is employed to achieve the expected complexity by adjusting the thresholds of a CU decision termination method. A rate-complexity-distortion model was suggested based on a large number of experiments [27], [28]. Different parameters have been tested to establish the model, which can quantify the relationship between the encoding efficiency and complexity. In this way, the expected complexity can be achieved by adjusting the parameters. Unfortunately, the complexity reduction fluctuates greatly as the video contents change. Most recently, Zhang *et al.* [29] proposed a CTU-level complexity control method that adaptively changes the partition depth for each CTU. A CTU-level complexity estimation and mapping model was introduced to improve the complexity control accuracy. Then, the complexity budgets are allocated to each CTU according to its estimated complexity. Moreover, Zhang *et al.* [30] subsequently proposed a complexity control strategy for HEVC intra coding, in which the framework is similar to the method in [29].

Different from existing algorithms, we adopt an online machine-learning method to overcome the complexity control problem in video coding by reasonably adjusting the accuracy of the CU decision model. To achieve better control accuracy, the proposed method adopts a multi-stage complexity allocation strategy to allocate the complexity budgets to each level more reasonably. In order to effectively and flexibly achieve a tunable trade-off between the RD performance and encoding complexity, a CTU-level complexity control scheme is well designed. Moreover, a feedback mechanism is introduced to eliminate the control error caused by improper complexity allocation.

III. Multi-Accuracy CU Decision Model

In fact, an effective complexity reduction strategy is the precondition for achieving complexity control in video coding [24]. According to the analyses in Section II-A, the brute-force RDO search-based CU decision is the main source of encoding complexity in HEVC inter coding. Therefore, we reduce the complexity by replacing the RDO search with a multi-accuracy CU decision model to determine the optimal CU depth. This model jointly utilizes an early skip model (ES) and an early termination model (ET) for efficient CU decision. It is noteworthy that the design of multi-accuracy CU decision aims to make the complexity reduction controllable.

A. Feature Extraction for CU Decision Model

Feature extraction plays an important role in the performance of classifiers, because they represent the attributes to distinguish different categories. In addition, the computational complexity of feature extraction should be taken into consideration when the classifier is introduced into a live video coding system. The CU decision in HEVC inter coding mainly depends on the texture of video content, motion and the context of spatiotemporal neighboring information. Therefore, we select 11 features for the proposed CU decision model.

1) Motion Information: As the CU decision in HEVC inter coding highly depends on the motion of video content, two motion features are introduced to train the CU decision model. Specifically, the motion vectors of Inter $2N \times 2N$ (denoted as x_{MV}) is selected for the ET classifier, and the Hash value (denoted as x_{Hash}) is used to train the ES classifier. Note that the Hash value is used to measure the similarity between the current CU and its co-located CU, which can indicate the motion difference in the temporal domain to some extent. The Hash value is defined as

$$H = \sum_{m=0}^{7}\sum_{n=0}^{7}\left\lceil \frac{1}{64} \times \left\lfloor \frac{f(m,n)_{Cur}}{\overline{f}_{Cur}} \right\rfloor \cdot \left\lfloor \frac{f(m,n)_{Col}}{\overline{f}_{Col}} \right\rfloor \right\rceil, \quad (2)$$

where $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ represent upward and downward rounding, respectively. $f(m,n)_{Cur}$ and $f(m,n)_{Col}$ represent the pixel value at (m,n) of the current CU and its co-located CU after being sampled to 8×8 respectively. Moreover, $\overline{f}_{Cur}$ and $\overline{f}_{Col}$ indicate the pixel mean of $f(m,n)_{Cur}$ and $f(m,n)_{Col}$, respectively.

2) Texture Information: The HEVC encoder achieves high encoding efficiency by encoding the textured regions using a smaller CU size, whereas the smooth regions are encoded with a larger CU size. In this work, texture complexity (denoted as x_{TC}), direction complexity (denoted as x_{DC}), and texture divergence between sub-CUs (denoted as x_{SubCU}) are selected as features. Specifically, the texture complexity of each CU is defined as

$$TC(\mathbf{\Omega}) = \sum_{(i,j)\in\mathbf{\Omega}} f(i,j)^2 - \frac{1}{N_\Omega}\Big[\sum_{(i,j)\in\mathbf{\Omega}} f(i,j)\Big]^2, \quad (3)$$

where $\mathbf{\Omega}$ is the block of pixels of the current CU, $N_{\mathbf{\Omega}}$ is the number of pixels in $\mathbf{\Omega}$, and $f(i,j)$ is the luminance component of the pixel at (i,j). The direction complexity is defined as

$$DC(\mathbf{\Omega}) = \frac{1}{N_{\mathbf{\Omega}}}\sum_{(i,j)\in\mathbf{\Omega}} \begin{matrix}(|G_{Hor}(i,j)| + |G_{Ver}(i,j)| \\ + |G_{45}(i,j)| + |G_{135}(i,j)|),\end{matrix} \quad (4)$$

where $G_n(i,j)$ is the Sobel gradient and is calculated by

$$G_n(i,j) = \mathbf{S}_n * \mathbf{F}, (n = Hor, Ver, 45°, 135°), \quad (5)$$

where $\mathbf{S}_n$ represents the four angular Sobel operators for the pixel at (i,j), and $\mathbf{F}$ is the 3×3 pixel matrix centered at (i,j).

Moreover, the texture divergence is obtained by

$$\mathcal{D}(\mathbf{\Omega}) = \frac{1}{N_{subCU}}\sum_{\mathbf{C}_i\in\mathbf{\Omega}}\left(TC(\mathbf{C}_i) - \frac{1}{N_{subCU}}\sum_{\mathbf{C}_i\in\mathbf{\Omega}} TC(\mathbf{C}_i)\right)^2 \quad i = \{1,2,3,4\}, \quad (6)$$

where $\mathbf{C}_i$ is the four sub-CUs of the current CU, and N_{subCU} is the number of sub-CUs of the current CU, which is equal to 4.

3) Coding Information of the Current CU: In order to exploit the features highly related with the CU decision, two stage CU decision models in this work (ES and ET models) are conducted after the checking Skip/Merge mode. Therefore, the coding information can be utilized to improve the coding performance. These selected features contain the RD cost of Skip/Merge, coding bits of Skip/Merge, coding flag of Skip, and coded block flag. They are denoted as x_{RDCost},

TABLE I
PERFORMANCE COMPARISON BETWEEN COMMON CLASSIFIERS

Classifier	Sequences	PA (%)	TT (s)
SVM	BQSquare	76.02	4.42
	PartyScene	67.13	23.48
	ChinaSpeed	84.28	47.26
	ParkScene	87.76	123.38
BPNN	BQSquare	76.66	0.93
	PartyScene	71.46	3.19
	ChinaSpeed	76.50	4.45
	ParkScene	79.20	0.01
kNN	BQSquare	72.36	0.01
	PartyScene	63.18	0.01
	ChinaSpeed	81.84	0.01
	ParkScene	86.03	0.01
Bayesian	BQSquare	76.84	0.01
	PartyScene	69.87	0.01
	ChinaSpeed	81.27	0.01
	ParkScene	81.25	0.01
RF	BQSquare	78.68	1.35
	PartyScene	71.12	5.68
	ChinaSpeed	86.08	10.03
	ParkScene	88.01	26.61

x_{Bits}, $x_{SkipFlag}$ and x_{CBF}, respectively. Generally, CU tends to select non-split as the best, when x_{RDCost} and x_{Bits} are small. $x_{SkipFlag}$ and x_{CBF} are also highly related with the CU decision, because they indicate the coding performance of the current depth.

4) Context Information: There is highly temporal correlation between the current frame and its reference frame, thus we select two temporal features. The first is the difference between the current CU depth and the maximum partition depth of its co-located CTU (denoted as $x_{\Delta Depth}$). The second is the difference of the QP value between the current frame and the reference frame in position 0 of the reference list 0 (denoted as $x_{\Delta QP}$). These two features can indicate the similarity between the current CU and its co-located CU, and the HEVC encoder tends to select depth of the co-located CU as the best depth of the current CU when $x_{\Delta Depth}$ and $x_{\Delta QP}$ are lower than a certain threshold.

B. Classifier Selection for CU Decision Model

Various classifiers have been introduced into CU decision model of HEVC, such as support vector machine (SVM), Bayesian method and back propagation neural network (BPNN). As we know, the prediction accuracy is a key factor for the classifier selection. In addition, the time overhead of training should also be taken into consideration, since the proposed model is obtained by online learning for accommodating different characteristics of input videos. In order to select the optimal classifier for this task, we have compared the prediction accuracy and online training speed of different classifiers, including SVM, BPNN, k-Nearest Neighbor (kNN), Bayesian and Random Forest (RF) [32]. Table I lists the experimental results, in which PA and TT indicate prediction accuracy and training time of classifier, respectively. Obviously, RF and SVM have good performance in terms of prediction accuracy. Moreover, the complexity overhead of model training of RF is significantly less than that of SVM. In addition, deep learning-based classifiers cannot meet the speed requirement of online learning task. Thus, RF is selected in this work by comprehensive consideration of both prediction accuracy and complexity overhead.

C. Partition Function Optimization for CU Decision Model

In this work, we exploit an effective partition function by learning the representative features in a randomized tree to generalize the rules of the HEVC encoder. Generally, random forest $\mathbf{T}=\{T_n\}_{t=1}^{T}$ is an ensemble classifier constructed by T binary random decision trees T_n [10]. Each random decision tree recursively separates the sample into child nodes on the left or right until a leaf node corresponding to the posterior distribution over the classes is reached. Given a node ϕ_i and its sample set $D_i = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$ where $\{\mathbf{x}_i, y_i\}$ is the i-th sample, $\mathbf{x}_i \in \Re^d$ is the feature vector and y_i is the label. The node needs to make an optimal binary decision, achieved by a partition function:

$$h_\psi(\mathbf{x}, \psi_i) : \Re^d \times \Psi \rightarrow \{0, 1\} \tag{7}$$

where ψ_i is the partition parameters of the node ϕ_i, and Ψ is the set of all partition parameters. Moreover, the outputs 1 and 0 represent the child nodes on the right and the left to be selected after partitioning, respectively.

In order to further improve the performance of the proposed CU decision model, the partition of the node ϕ_i is measured by the Gini coefficient [33], which is calculated as

$$\begin{aligned} G(D_i) &= \sum_{k=1}^{K} p_{i,k}(1 - p_{i,k}) = 1 - \sum_{k=1}^{K} p_{i,k}^2 \\ &= 1 - \sum_{k=1}^{K} \left(\frac{N_{i,k}}{N_i}\right)^2, \end{aligned} \tag{8}$$

where N_i is the number of elements of D_i, $p_{i,k}$ is the probability of elements belonging to class k, and $N_{i,k}$ is the number of elements belonging to class k in D_i. Moreover, K is 2 in the binary decision. Lastly, the best partition of the node ϕ_i is determined by the Gini coefficient of a feature Θ when D_i is divided into two subsets D_i^L and D_i^R in the child note on the left and the right. It can be represented as

$$G_{Split}(D_i, {\Theta_i}^{\xi}) = \frac{N_i^L}{N_i} G(D_i^L) + \frac{N_i^R}{N_i} G(D_i^R) \tag{9}$$

where ξ is a threshold determined by learning, N_i^L and N_i^R are the numbers of elements of D_i^L and D_i^R, respectively. Based on (8), (9) can be rewritten as

$$\begin{aligned} &G_{Split}(D_i, {\Theta_i}^{\xi}) \\ &= \frac{N_i^L}{N_i}[1 - \sum_{k=1}^{K}\left(\frac{N_{i,k}^L}{N_i^L}\right)^2] + \frac{N_i^R}{N_i}[1 - \sum_{i=1}^{m}\left(\frac{N_{i,k}^R}{N_i^R}\right)^2] \\ &= 1 - \frac{1}{N_i}\left(\frac{1}{N_i^L} \times \sum_{k=1}^{K}\left(N_{i,k}^L\right)^2 + \frac{1}{N_i^R} \times \sum_{k=1}^{K}\left(N_{i,k}^R\right)^2\right) \end{aligned} \tag{10}$$

where $N_{i,k}^L$ and $N_{i,k}^R$ are the numbers of elements belonging to class k in D_i^L and D_i^R, respectively.

In this work, the CU decision model is obtained by an online learning approach to accommodate the different characteristics of video sequences. However, the proportion of positive and

negative samples in the training set obtained by online collecting is extremely uneven, and seriously affects the encoding performance of the CU decision model. Consequently, we introduce the class weight (denoted as W_k) to enhance the encoding performance of the CU decision model. The class weight is utilized to improve the prediction accuracy of the class with smaller samples by specifying more severe penalties for the misclassification of small class samples. Considering the difference in the number of samples between classes, the class weight of the node ϕ_i is normalized as

$$\delta_{i,k} = \frac{W_k}{N_{i,k}} \Big/ \sum_{k=1}^{K} \frac{W_k}{N_{i,k}} \tag{11}$$

Then, combining (10) with (11), the partition function can be represented as

$$G_{Split}(D_i, \Theta^{\xi}, \boldsymbol{\delta}) = 1 - \frac{1}{N_i}\left(\frac{1}{N_i^L}\sum_{k=1}^{K}\left(\delta_{i,k}\cdot N_{i,k}^L\right)^2 + \frac{1}{N_i^R}\sum_{k=1}^{K}\left(\delta_{i,k}\cdot N_{i,k}^R\right)^2\right). \tag{12}$$

Moreover, according to (11), N_i^L and N_i^R can be transformed as

$$\begin{cases} N_i^L = \sum_{k=1}^{K} \delta_{i,k} \times N_{i,k}^L \\ N_i^R = \sum_{k=1}^{K} \delta_{i,k} \times N_{i,k}^R. \end{cases} \tag{13}$$

Thus, the function can be rewritten as

$$G_{Split}(D_i, \Theta^{\xi}, \boldsymbol{\delta}) = 1 - \frac{1}{N_i}\left(\frac{\sum_{i=1}^{K}\left(\delta_{i,k}\times N_{i,k}^L\right)^2}{\sum_{i=1}^{K}\delta_{i,k}\times N_{i,k}^L} + \frac{\sum_{i=1}^{K}\left(\delta_{i,k}\times N_{i,k}^R\right)^2}{\sum_{i=1}^{K}\delta_{i,k}\times N_{i,k}^R}\right), \tag{14}$$

where $\boldsymbol{\delta} = (\delta_{i,0}, \delta_{i,1})$ is the set of class weights. In this way, the encoding performance of the proposed random forest-based CU decision model can be adjusted by changing the class weight.

In the prediction stage, each internal node randomly selects a fixed number of features and their corresponding thresholds. In order to balance the encoding performance and computational complexity, the number of features of each node is set as 3. The decision criterion of the optimal leaf node is represented as

$$y^* = \arg\min_{y_i \in \mathbf{Y}}(G_{Split}(D_i, \Theta^{\xi}, \boldsymbol{\delta})). \tag{15}$$

It is noteworthy that the final decision is made by obtaining the results of all the trees, i.e., by majority voting.

D. Multi-Accuracy Optimization for CU Decision Model

Because samples with different categories usually overlap with each other, it is difficult to find a perfect classifier to distinguish these two categories without misclassification. Basically, there are two kinds of misclassification: 1) the CUs belonging to Class 0 are misclassified into Class 1, and 2) CUs belonging to Class 1 are misclassified into Class 0. Theoretically, the RDO search of the CUs belonging to Class 0 should be early terminated/skipped in the proposed CU decision model. However, the RDO search will be not terminated/skipped duo to the first kind of misclassification. Consequently, the amount of encoding time saving is reduced. It should be noted that the first kind of misclassification does not result in any RD performance loss because the full RDO search is conducted, which is consistent with the original HEVC encoder. For the second kind of misclassification, the RDO search of these misclassified CUs is early terminated by the ET model or early skipped by the ES model. Therefore, it degrades the RD performance. Based on the above analysis, the correct classification of CUs belonging to Class 0 is beneficial to achieve further complexity reduction, whereas the correct classification of CUs in Class 1 can maintain the RD performance. In order to measure the classification performance of the CU decision model, we define the prediction accuracies of Class 0 and Class 1 (denoted as PA_0 and PA_1, respectively)

$$PA_0 = \frac{K_{00}}{K_{01} + K_{00}}, \quad PA_1 = \frac{K_{11}}{K_{11} + K_{10}} \tag{16}$$

where K_{00} is the number of CUs correctly classified as Class 0, K_{01} is the number of the first misclassified CUs, K_{10} is the number of the second misclassified CUs, and K_{11} is the number of CUs correctly classified as Class 1.

Further, extensive experiments have been conducted to quantify the influence of the class weights (W_0 and W_1) on the proposed CU decision models at each depth level. In these experiments, W_0/W_1 ranges from 1:10 to 10:1, and the step is 1. The relationship between the class weights and prediction accuracies of models at different depth levels are shown in Fig. 7. As W_0/W_1 increases, the penalty for the misclassification of CUs belonging to Class 0 increases; thus, PA_0 of each model increases, whereas PA_1 decreases. This is consistent with the aforementioned analysis. Based on the relationship between the class weight and prediction accuracy of the proposed CU decision model, we can obtain three models with different prediction accuracies (high, medium, and low accuracy). Specifically, the high-accuracy CU decision model achieves smaller RD performance loss at the cost of lower complexity reduction. In contrast, the CU decision model with low prediction accuracy reduces the complexity to a larger extent, but the RD performance loss is larger. In addition, the CU decision model with medium prediction accuracy maintains a more acceptable trade-off between complexity reduction and RD performance. The selected class weights for the CU decision models with different accuracies and their prediction accuracies are listed in Table II. Based on the design of multi-accuracy CU decision, CTU-level encoding complexity control can be achieved by reasonably selecting the prediction accuracy of the CU decision model.

IV. Multi-Stage Complexity Control Scheme

In order to achieve accurate sequence-level complexity control, the proposed complexity control method contains two stages: multi-stage complexity allocation and CTU-level complexity control. The coding structure of the proposed

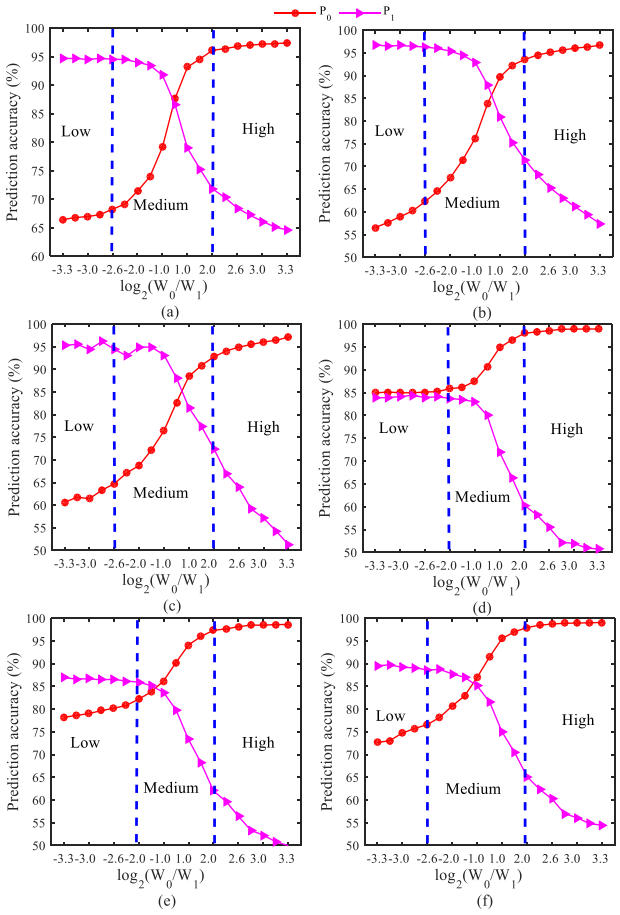


Fig. 7. Prediction accuracy of different classifiers. (a) ES(0) (b) ES(1) (c) ES(2) (d) ET(0) (e) ET(1) (f) ET(2).

TABLE II
SELECTED PARAMETERS FOR CLASSIFIERS

Depth	Accuracy	ES		ET	
		W_0/W_1	PA_0	W_0/W_1	PA_0
0	High	4:1	96.3%	4:1	98.2%
	Medium	1:1	88.0%	1:1	90.7%
	Low	1:6	68.1%	1:4	85.5%
1	High	4:1	94.4%	4:1	98.0%
	Medium	1:1	87.6%	1:1	90.3%
	Low	1:6	62.3%	1:4	83.2%
2	High	4:1	93.9%	4:1	98.6%
	Medium	1:1	83.2%	1:1	91.5%
	Low	1:6	64.9%	1:6	76.5%

method is shown in Fig. 8. We define all frames between two control model updating frames as a segment. For an input video sequence, training frames (the first 12 P frames) are coded by original HEVC encoder for collecting the training data and online training the multi-accuracy CU decision model. After obtaining the multi-accuracy CU decision model, the first frames of each segment (updating frames) are also coded by original encoder for updating the parameters of multi-stage complexity control model. Finally, the constrained frames are coded by the multi-accuracy CU decision model to

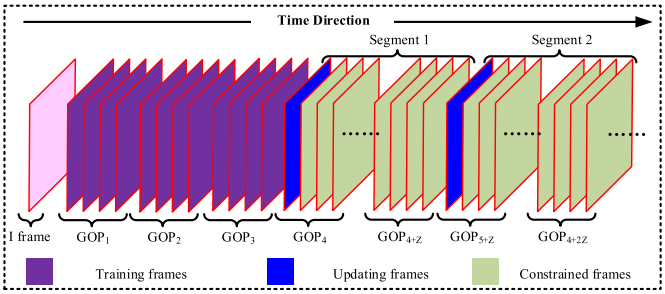


Fig. 8. Coding structure of the proposed method.

reduce the total encoding complexity to approach the expected complexity. In addition, a feedback mechanism is used to eliminate the complexity control error caused by inappropriate complexity allocation.

A. Multi-Stage Complexity Allocation

In this article, complexity allocation is implemented on four levels: segment, GOP, frame, and CTU.

1) Segment-Level Complexity Allocation: Based on the analysis in Section II-A, each GOP in a sequence consumes approximately the same encoding time when there is not the scene content change. Meanwhile, the encoding complexity of a GOP can be derived by a certain frame and its complexity ratio. In order to achieve accurate complexity control, the first frame in each segment is used to update the estimated complexity of the current segment. Therefore, the encoding complexity of the current segment can be calculated as

$$T_s^{Seg} = Z \times T_{s,g}^{GOP} = Z \times \frac{T_{s,g,1}^{Frame}}{\beta_{s,g,1}^{Frame}} \tag{17}$$

where Z is the number of GOPs in a segment, $T_{s,g}^{GOP}$ is the encoding time of the first GOP in the current segment, $T_{s,g,1}^{Frame}$ is the encoding time of the first frame in the current segment, and $\beta_{s,g,1}^{Frame}$ is the complexity ratio of the first frame. Mathematically, the encoding complexity of the entire sequence can be estimated as

$$\begin{aligned} T_{ES}^{Seq} &= \sum_{f=1}^{13} T_f^{TF} + \sum_{s=1}^{S} T_s^{Seg} \\ &= \underbrace{\sum_{f=1}^{13} T_f^{TF}}_{\text{Complexity of TF}} + \underbrace{\sum_{s=1}^{S} Z \times \frac{T_{s,g,f}^{Frame}}{\beta_{s,g,1}^{Frame}}}_{\text{Complexity of S segments}} . \end{aligned} \tag{18}$$

In this article, we adopt a complexity allocation strategy that if the previous segments consume more or fewer complexity than the expected complexity, then the current segment should be allocated fewer or more complexity accordingly. The expected complexity for the sth segment is calculated by

$$\overline{T}_s^{Seg} = \frac{1}{S-s+1} \cdot \left(T_{ES}^{Seq} \cdot r_{target} - \sum_{f=1}^{13} T_f^{TF} - \sum_{x=1}^{s-1} T_x^{Seg}\right), \tag{19}$$

where r_{target} is the expected complexity ratio, T_f^{TF} is the encoding time of training frames, and S is the number of segments.

2) GOP-Level Complexity Allocation: Based on the statistical analysis in Section II-A, we can allocate the leftover complexity budgets to the remaining GOPs in the current segment by averaging. Thus, GOP-level expected complexity is calculated by

$$\overline{T}_z^{GOP} = \frac{1}{Z-z+1} \times (\overline{T}_s^{Seg} - T_{Coded}^{Seg}), \tag{20}$$

where $\overline{T}_z^{GOP}$ is the expected complexity of the z-th GOP in the current segment, and T_{Coded}^{Seg} is the encoding time of the previous $z-1$ GOPs in the current segment.

3) Frame-Level Complexity Allocation: The frame-level expected complexity can be determined by the complexity ratio, since the frame complexity ratios of different sequences with same QP are approximately same according to the statistical analysis in Section II-A. Mathematically, it can be calculated as

$$\overline{T}_r^F = \frac{\overline{T}_z^{GOP} - T_{coded}^{GOP}}{\sum_{\{\,ALL\ Not\ Coded\ CTUs\}} \beta_r^{Frame}} \times \beta_r^{Frame}, \tag{21}$$

which aims to allocate the leftover complexity budgets to the remaining frames according to the weight of each frame.

4) CTU-Level Complexity Allocation: Similar to frame-level complexity allocation, the CTU-level complexity also aims to allocate the leftover complexity budgets to the remaining CTUs according to the weight of each CTU. The target CTU-level complexity is calculated as

$$\overline{T}_n^{CTU} = \frac{\overline{T}_r^F - T_{Coded}^F}{\sum_{\{\,ALL\ Not\ Coded\ CTUs\}} w_n^{CTU}} \times w_n^{CTU}, \tag{22}$$

where T_{Coded}^F is the complexity of coded CTUs in the current frame, and w_n^{CTU} is the weight of the current CTU.

Algorithm 1 Multi-Stage Complexity Control

Input: Input video and target complexity r_{target}
Output: Appropriate CU decision model for each CTU

```
Initialize S as the total number of segments in a sequence;
Initialize Z as the total number of GOPs in a segment;
Initialize N as the total number of CTUs in a frame;
for f = 1; f ≤ 13; f++ do
    Record the encoding time of the first 13 frames;
    Collect the training data;
end
Train the multi-accuracy CU decision model;
Estimate the total encoding time T_ES^Seq using (18);
for s = 1; s ≤ S; s++ do
    Calculate the segment-level target complexity T̄_s^Seg using (19);
    for z = 1; z ≤ Z; z++ do
        Calculate the GOP-level target complexity T̄_z^GOP using (20);
        for r = 1; r ≤ 4; r++ do
            Calculate the frame-level target complexity T̄_r^F using (21);
            for n = 1; n ≤ N; n++ do
                Calculate the CTU-level target complexity T̄_n^CTU using (22);
                Select CU decision model using (24) and (25);
            end
        end
    end
end
```

B. CTU-Level Complexity Control

In Section IV-A, we introduced an approach for allocating complexity budgets to the four coding levels. However, we still cannot accurately code each sequence under the expected complexity if we only know the allocated complexity. The aim of CTU-level complexity control is to reduce the complexity of each frame to achieve the expected complexity, which is based on the multi-accuracy CU decision model introduced in Section III. The complexity control strategy can be expressed by

$$\arg\min_{\{\mathbf{A}\}} \left| \sum_{n=1}^{N} T_n^{CTU} - \overline{T}_r^F \right|, \tag{23}$$

where N is the number of CTUs in each frame, T_n^{CTU} is the actual encoding time of the n-th CTU in the current frame, and $\mathbf{A}$ is the set of accuracy of the CU decision models used to encode different CUs in a frame. To solve (23), we define the complexity control error as

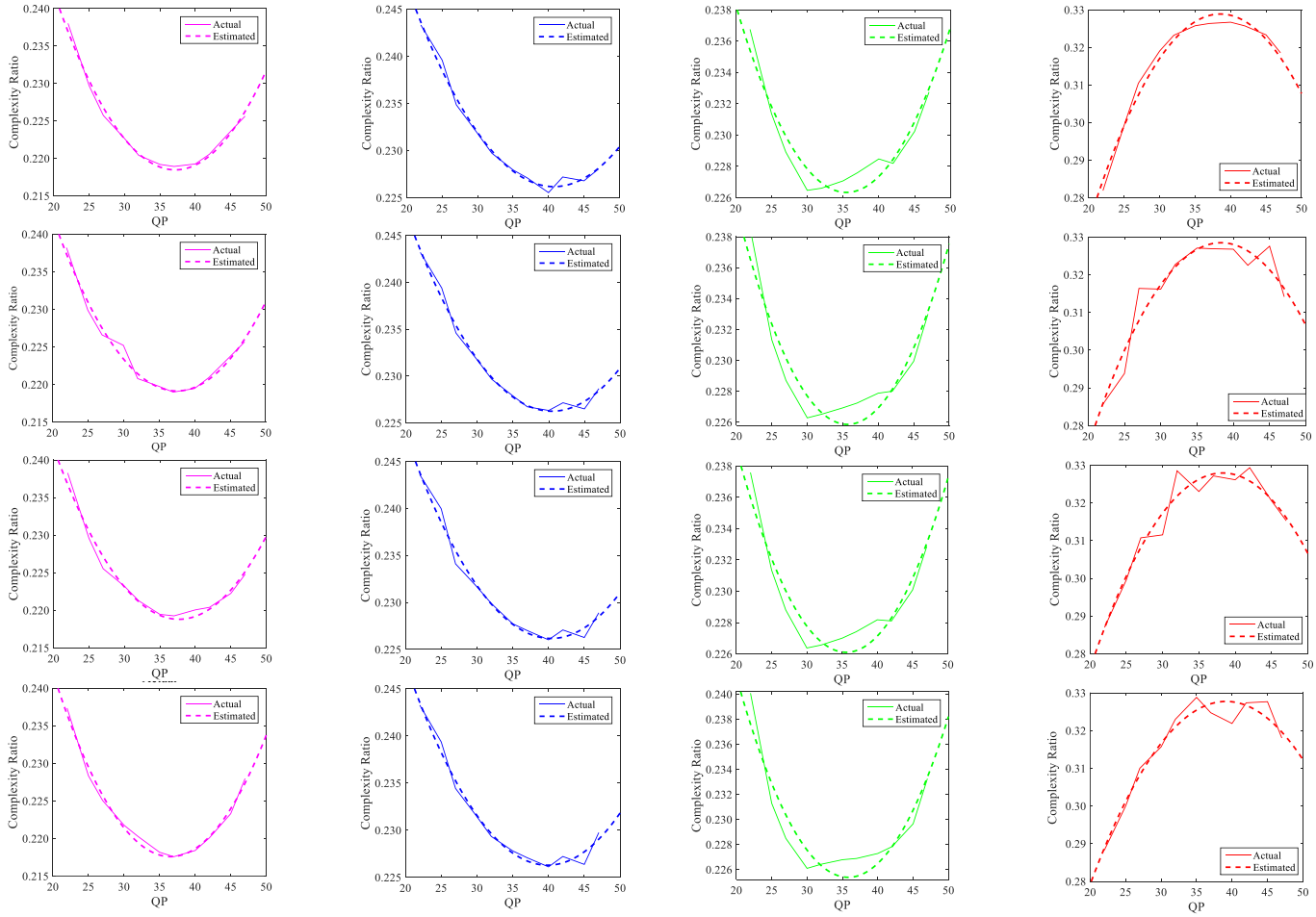
$$\begin{aligned} e_n &= \sum_{n=1}^{N} \overline{T}_n^{CTU} - \sum_{n=1}^{N} T_n^{CTU} \\ &= \sum_{n=1}^{N} (\overline{T}_r^F \times w_n^{CTU} \Big/ \sum_{n=1}^{N} w_n^{CTU}) - \sum_{n=1}^{N} T_n^{CTU}, \end{aligned} \tag{24}$$

where e_n indicates sum of the control errors caused by the previous N CTUs in the current frame.

Consequently, the complexity control is transformed to solve the problem of eliminating the controlling error e_n. In this article, we eliminate the e_n by selecting the prediction accuracy of the CU decision model. The strategy we adopt determines that, if the previous CTUs consume more or less complexity than the expected complexity, then the current CTU should consume less or more complexity accordingly. As introduced in Section III-D, the high-, medium- and low-accuracy models can be used to modulate the trade-off between the complexity reduction and RD performance. Therefore, the optimal accuracy selection in the proposed algorithm can be expressed as

$$A = \begin{cases} A_{Low} & e_n \le -MT \\ A_{Mid} & -MT < e_n \le 0 \\ A_{High} & 0 < e_n \le MT \\ A_{Ori} & MT < e_n \end{cases} \tag{25}$$

where A_{Low}, A_{Mid}, A_{High}, and A_{Ori} are the CU decision models with prediction accuracy of low, medium, high, and original, respectively. Moreover, MT is the average encoding time of CTUs in the training frames.



(1) β_1^{Frame} (2) β_2^{Frame} (3) β_3^{Frame} (4) β_4^{Frame}

Fig. 9. Estimated and actual frame-level complexity ratio under different QPs. From top to bottom: *RaceHorses*, *BQMall*, *ParkScene*, *PartyScene* From Left to right: β_1^{Frame}, β_2^{Frame}, β_3^{Frame}, β_4^{Frame}.

Overall, the parameters learning of the proposed multi-stage complexity control can be described as Algorithm 1.

C. Parameters Determination

1) Determination of β_r^{Frame}: Based on the statistical analysis in Section II-A, the complexity ratio β_r^{Frame} varies with the temporal layer in the LDHCS. Moreover, β_r^{Frame} is also related with QP. In order to determine the value of β_r^{Frame}, five sequences were encoded with different QPs under the low delay P main configuration. Fig. 9 shows each β_r^{Frame} and its estimated curve. Clearly, the relationship between each β_r^{Frame} and QP can be fitted with a parabolic curve, which is represented as

$$\beta_r^{Frame} = \alpha_r \times Q^2 + \varphi_r \times Q + \tau_r, r \in \{1, 2, 3, 4\}, \quad (26)$$

where α_r, φ_r, and τ_r are constants, and Q is the value of QP. Table III lists the values of these parameters.

2) Determination of w_n^{CTU}: The weight of each CTU is determined by the mean CTU partition depth (MD), which is calculated as

$$MD = \sum_{d=0}^{3} n_d \times \frac{d}{2^{d+1}}, \quad (27)$$

TABLE III
PARAMETERS FOR COMPLEXITY RATIO

β_r^{Frame}	Parameter	Seq1	Seq2	Seq3	Seq4	Seq5	**Average**
β_1^{Frame}	α_1	8.0×10^{-5}	7.6×10^{-5}	7.3×10^{-5}	8.9×10^{-5}	6.2×10^{-5}	**7.6×10^{-5}**
	φ_1	-6.0×10^{-3}	-5.6×10^{-3}	-5.5×10^{-3}	-6.5×10^{-3}	-4.8×10^{-3}	**-5.7×10^{-3}**
	τ_1	3.3×10^{-1}	3.3×10^{-1}	3.2×10^{-1}	3.4×10^{-1}	3.1×10^{-1}	**3.3×10^{-1}**
β_2^{Frame}	α_2	5.1×10^{-5}	5.2×10^{-5}	5.2×10^{-5}	5.4×10^{-5}	5.0×10^{-5}	**5.2×10^{-5}**
	φ_2	-4.1×10^{-3}	-4.1×10^{-3}	-4.2×10^{-3}	-4.3×10^{-3}	-4.0×10^{-3}	**-4.2×10^{-3}**
	τ_2	3.1×10^{-1}	3.1×10^{-1}	3.2×10^{-1}	3.1×10^{-1}	3.1×10^{-1}	**3.1×10^{-1}**
β_3^{Frame}	α_3	5.0×10^{-5}	5.7×10^{-5}	5.3×10^{-5}	6.4×10^{-5}	5.0×10^{-5}	**5.5×10^{-5}**
	φ_3	-3.5×10^{-3}	-4.1×10^{-3}	-3.8×10^{-3}	-4.6×10^{-3}	-3.5×10^{-3}	**-3.9×10^{-3}**
	τ_3	2.9×10^{-1}	3.0×10^{-1}	2.9×10^{-1}	3.1×10^{-1}	2.9×10^{-1}	**3.0×10^{-1}**
β_4^{Frame}	α_4	-1.6×10^{-4}	-1.6×10^{-4}	-1.6×10^{-4}	-1.3×10^{-4}	-1.6×10^{-4}	**-1.5×10^{-4}**
	φ_4	1.2×10^{-2}	1.2×10^{-2}	1.2×10^{-2}	1.0×10^{-2}	1.2×10^{-2}	**1.1×10^{-2}**
	τ_4	8.9×10^{-2}	9.2×10^{-2}	1.0×10^{-1}	1.3×10^{-1}	8.9×10^{-2}	**1.0×10^{-1}**

*1: RaceHorses, 2: BQMall, 3: ParkScene, 4: PartyScene, 5: Johnny.

where n_d is the number of CUs with depth of d in the current CTU. To accurately determine the relationship between the weight of each CTU and MD, the conditional probability $P(d|MD)$ is defined, where d represents the optimal depth of the current CTU. For instance, $P(d = 1|MD < 0.5)$ indicates

TABLE IV
THE $P(d|MD)$ FOR DIFFERENT VIDEOS

	$P(d\|MD)$	d=0	d=1	d=2	d=3
BQMall	$MD \leq 0.5$	**100**	0.00	0.00	0.00
	$0.5 < MD \leq 1.5$	0.00	**80.35**	**16.78**	2.87
	$1.5 < MD \leq 2.5$	0.00	**28.13**	**48.40**	**23.48**
	$2.5 < MD$	0.00	0.00	**35.52**	**64.48**
	$P(d\|MD)$	d=0	d=1	d=2	d=3
Johnny	$MD \leq 0.5$	**100**	0.00	0.00	0.00
	$0.5 < MD \leq 1.5$	0.00	**85.14**	**13.81**	1.05
	$1.5 < MD \leq 2.5$	0.00	**38.12**	**49.37**	**12.51**
	$2.5 < 2.5$	0.00	0.00	**37.74**	**62.26**
	$P(d\|MD)$	d=0	d=1	d=2	d=3
ParkScene	$MD \leq 0.5$	**100**	0.00	0.00	0.00
	$0.5 < MD \leq 1.5$	0.00	**81.58**	**16.51**	1.91
	$1.5 < MD \leq 2.5$	0.00	**27.54**	**51.78**	**20.67**
	$2.5 < MD$	0.00	0.00	**35.03**	**64.97**

the probability of the event that the optimal depth of the current CTU is 1 when its MD is less than 0.5. Table IV presents the $P(d|MD)$ for three different videos. Clearly, the CTUs with a smaller MD are highly probable to have a lower optimal depth d. For instance, when MD is less than 0.5, the CTU has a 100% probability of its optimal depth being 0 for three test sequences. In other words, the current CTU only needs to conduct the RDO search at a depth level of 0. Based on the statistics in Table IV, the weight is determined by

$$w_n^{CTU} = \begin{cases} CR_0 & MD \leq 0.5 \\ CR_1 + CR_2 & 0.5 < MD \leq 1.5 \\ CR_1 + CR_2 + CR_3 & 1.5 < MD \leq 2.5 \\ CR_2 + CR_3 & 2.5 < MD, \end{cases} \tag{28}$$

where CR_d represents the complexity ratio consumed by RDO search at depth level d. It can be calculated by

$$CR_d = T_d^{TF} \Big/ \sum_{d=0}^{3} T_d^{TF}, \tag{29}$$

where T_d^{TF} is the average encoding time consumed by RDO search at depth level d in the training frames. It should be noted that, because the depth information of the current CTU can only be available after coding, we utilize the mean CTU depth of the co-located CTU MD_{Col} to derive the MD of the current CTU. Thus, MD can be calculated as

$$\begin{cases} MD = MD_{Col} + \Delta MD \\ \Delta MD = MD_{STL} - MD_{RF}, \end{cases} \tag{30}$$

where MD_{STL} is the average depth of the frames at the same temporal layer with the current frame (STLF), and MD_{RF} is the average depth of the frames that are the reference frame of STLFs.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

This section presents the extensive experiments that were conducted to evaluate the performance of the proposed method. The proposed method was implemented on the HEVC test platform HM-16.0. All experiments were conducted under the comment test conditions (CTC) of JCT-VC. The low-delay P main configuration was used. The accuracy of the complexity control was evaluated by control error CE, which is calculated as

$$CE = \frac{1}{4} \sum_{i=1}^{4} \left| \frac{T_j^{QP_i}}{T_{HM}^{QP_i}} - r_{target} \right| \times 100\%, \tag{31}$$

where $T_{HM}^{QP_i}$ and $T_j^{QP_i}$ are the encoding time of the original HM and j-th algorithm, respectively. r_{target} is the expected complexity ratio. Moreover, the RD performance was evaluated by the Bjøntegaard Delta Peak Signal-to-Noise Ratio (BDPSNR) and Bjøntegaard Delta Bitrate (BDBR) [34].

B. Control Accuracy of the Proposed Method

The complexity control performance of the proposed method under different target complexities is listed in Table V. It can be observed that CE for different sequences under 80% expected complexity range from 0.03% to 4.43%, and 1.30% on average. When the expected complexity is set as 60%, CE for different sequences range from 0.36% to 3.38%, and 1.29% on average. Moreover, CE for different sequences range from 0.36% to 5.09%, and 1.69% on average. The statistical data show that CE of the proposed method is quite small, and consistent for different target complexities. In other words, the proposed complexity control method is accurate and stable for various sequences and different target complexities. Therefore, the HEVC encoder can be efficiently implemented on video-capable devices with different computational capacities and constrained power by using the proposed complexity control method. The multi-stage complexity allocation mechanism is largely responsible for the highly accurate complexity control, because it reasonably allocates the expected complexity to each coding level.

C. RD Performance of the Proposed Method

The RD performance of the proposed method under different target complexities is also provided in Table V. The average values of BDBR and BDPSNR are 0.41% and -0.018dB, respectively, when the expected complexity is 80%. The average values of BDBR and BDPSNR are 1.27% and -0.043dB, respectively, when the expected complexity is 60%. Furthermore, when the expected complexity is 40%, the average values of BDBR and BDPSNR are 4.85% and -0.192dB, respectively. Clearly, when the expected complexity is 40%, the RD performance of the proposed method is more appropriate for high-resolution videos with more static and simple texture regions (such as *Johnny*, *Vidyo1* and *Vidyo3*). By contrast, when the low-resolution videos contain more motion and complexity texture regions (such as *BasketballPass* and *BasketballDrill*), the RD performance of the proposed method decreases slightly. The reason is that the redundant encoding complexities caused by unnecessary CU checking in videos with lower resolution and abundant motions are less than that in videos with higher resolution and simple motions. To achieve 40% expected complexity, the proposed method should select the CU decision models with low prediction accuracy to increase the complexity reduction, which leads to more RD performance loss. Theoretically, this can be

TABLE V
PERFORMANCE EVALUATION OF THE PROPOSED METHOD FOR DIFFERENT TARGET COMPLEXITY

Sequences		r_{target}=80%			r_{target}=60%			r_{target}=40%		
		CE (%)	BDPSNR (dB)	BDBR (%)	CE (%)	BDPSNR (dB)	BDBR (%)	CE (%)	BDPSNR (dB)	BDBR (%)
Class B	BasketballDrive	1.01	-0.016	0.65	1.71	-0.045	1.85	1.23	-0.124	5.39
	BQTerrace	0.09	-0.050	0.29	0.37	-0.017	0.98	0.95	-0.048	2.91
	ParkScene*	1.45	-0.014	0.45	2.17	-0.044	1.41	0.66	-0.104	3.43
	Kimono1	3.59	-0.031	0.96	3.38	0.069	2.17	2.83	-0.159	4.97
	Cactus	1.95	-0.013	0.57	1.65	-0.041	1.88	1.73	-0.141	6.48
Class C	PartyScene*	2.20	-0.021	0.49	1.31	-0.065	1.58	0.72	-0.267	6.57
	RaceHorsesC	1.62	-0.018	0.47	2.16	-0.050	1.31	1.54	-0.284	7.70
	BasketballDrill	1.62	-0.022	0.56	0.77	-0.055	1.41	1.27	-0.302	8.18
	BQMall	4.43	-0.029	0.76	3.01	-0.093	2.45	3.52	-0.304	8.08
Class D	BasketballPass	1.02	-0.034	0.72	0.51	-0.104	2.22	2.57	-0.414	9.22
	BQSquare	1.22	-0.010	0.27	0.51	-0.042	1.10	1.38	-0.211	5.81
	BlowingBubbles	2.46	-0.013	0.34	1.22	-0.104	2.76	2.05	-0.232	6.27
	RaceHorses*	2.04	-0.025	0.55	3.33	-0.083	1.82	5.09	-0.346	7.82
Class E	Johnny*	0.03	-0.001	0.10	0.54	-0.007	0.40	0.65	-0.032	1.46
	FourPeople	0.54	-0.005	0.15	1.34	-0.013	0.41	0.43	-0.076	2.32
	Vidyo1	0.44	-0.001	0.06	0.33	-0.005	0.16	0.33	-0.045	1.61
	Vidyo3	0.59	-0.012	0.45	0.72	-0.021	0.73	1.64	-0.055	1.93
	Vidyo4	0.08	-0.000	-0.02	0.36	-0.008	0.30	1.69	-0.073	2.72
	KristenAndSara	0.02	-0.004	0.12	0.10	-0.007	0.21	2.05	-0.099	3.40
Class F	SlideEditing	1.31	0.002	-0.01	0.38	-0.023	0.15	0.20	-0.048	0.32
	SlideShow	0.05	-0.023	0.29	0.81	-0.085	1.10	0.90	-0.407	5.29
	ChinaSpeed*	1.62	-0.042	0.78	2.21	-0.094	1.75	1.46	-0.355	6.82
	BaskeballDrillTe	0.61	-0.020	0.50	0.75	-0.067	1.67	3.95	-0.287	7.24
Average exclude *		**1.26**	**-0.017**	**0.40**	**1.12**	**-0.040**	**1.27**	**1.68**	**-0.183**	**4.99**
Average overall		**1.30**	**-0.017**	**0.41**	**1.29**	**-0.044**	**1.30**	**1.69**	**-0.192**	**5.04**

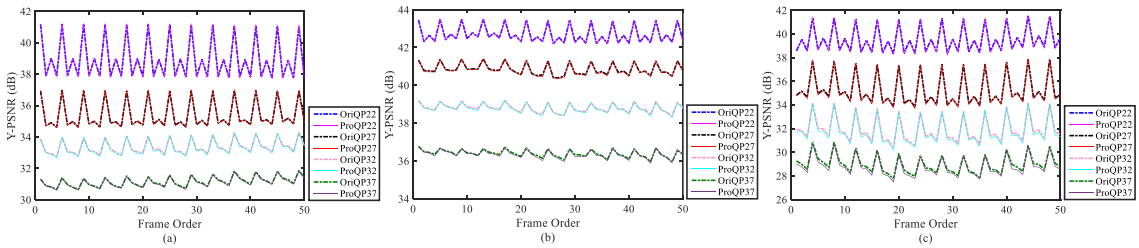


Fig. 10. Frame-level objective quality evaluation for the proposed method. (a) *BQTerrace* (r_{target} = 80%) (b) *Johnny* (r_{target} = 60%) (c) *RaceHorses* (r_{target} = 40%).

explained by the purpose of complexity control. Complexity control aims to achieve a trade-off between RD performance and encoding complexity flexibly and efficiently. In this regard, encoding complexity can be controlled accurately in a wide range along with expected RD performance degradation.

D. Frame-Level RD Performance of the Proposed Method

In addition to the above two properties, the frame-level quality and bit consumption of the proposed method are also evaluated. Fig. 10 shows the frame-level objective quality comparison for three sequences between the proposed method and the original HEVC encoder. The frame-level video quality curves of the proposed method are consistent with those of the original encoder. It indicates that the proposed method can efficiently reduce the encoding complexity to approach the target while maintaining the same video quality. A comparison of the frame-level bitrate for another three sequences between the proposed method and original HEVC platform is shown in Fig. 11. Similarly, the frame-level bitrate curves of the proposed method basically coincide with those of the original encoder, which indicates the increase in the bitrate consumption of our method is negligible. The excellent frame-level RD performance of the proposed method benefits from the frame-level complexity allocation mechanism, which reasonably allocates the complexity budgets according to the frame-level weight.

E. Comparison of Control Accuracy

The control accuracy of the proposed method is compared with three state-of-the-art complexity control algorithms, namely Deng-ACCESS [25], Deng-TCSVT [24], and Amaya-TMM [26]. Three target complexity ratios (80%, 60%, and 40%), which are same as those in the three related studies, were tested to ensure a fair comparison. In addition, the standard deviation (denoted as SD) is utilized to evaluate the fluctuation of properties for different methods. The experimental results are listed in Tables VI-VIII. Clearly, CE and SD of the proposed method for all expected complexity ratios are

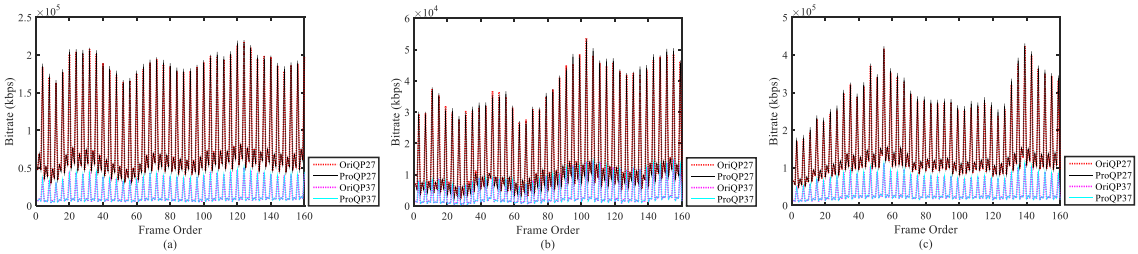


Fig. 11. Frame-level bitrate evaluation for the proposed method. (a) *PartyScene* (r_{target} = 80%) (b) *Vidyo3* (r_{target} = 60%) (c) *BasketballDrive* (r_{target} = 40%).

TABLE VI

PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND STATE-OF-THE-ART ALGORITHMS WITH 80% EXPECTED COMPLEXITY

Sequences	Deng_ACCESS			Deng_TCSVT			Amaya_TMM			Proposed		
	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)
BasketballDrive	1.42	-0.03	0.33	5.75	-0.05	0.41	2.9	**-0.01**	0.32	**1.01**	**-0.01**	**0.09**
BQTerrace	0.25	-0.02	0.33	4.81	-0.04	0.52	0.6	-0.02	0.59	**0.09**	**-0.01**	**-0.56**
ParkScene	**0.00**	-0.03	0.60	4.23	-0.05	0.73	0.49	-0.03	0.57	1.45	**-0.01**	**-0.02**
Johnny	0.84	**-0.00**	-0.05	0.71	-0.01	0.05	0.65	-0.00	0.08	**0.03**	-0.01	**-0.50**
FourPeople	0.93	**-0.00**	0.13	1.32	-0.03	0.23	0.93	-0.01	0.16	**0.54**	-0.02	**-0.40**
Vidyo1	1.98	**-0.00**	0.03	0.34	-0.02	0.10	1.39	-0.01	0.24	**0.33**	-0.01	**-0.23**
Vidyo3	1.11	**-0.00**	**0.04**	1.33	-0.03	0.14	6.07	-0.00	-0.17	**0.59**	-0.01	0.09
Vidyo4	2.28	**-0.00**	0.15	2.49	-0.02	0.11	1.93	-0.02	0.07	**0.08**	-0.01	**0.04**
PartyScene	1.27	-0.10	2.53	2.14	-0.20	3.12	**1.18**	-0.17	3.62	2.20	**-0.01**	**0.42**
RaceHorses	1.69	-0.08	1.56	**0.03**	-0.12	2.30	6.23	-0.05	0.92	1.62	**-0.00**	**0.31**
BasketballDrill	2.75	-0.02	0.46	2.81	-0.15	2.32	2.94	-0.02	0.79	**1.62**	**-0.01**	**0.18**
SlideEditing	1.53	-0.03	0.42	3.46	-0.07	0.98	2.34	**-0.02**	0.23	**1.31**	**-0.02**	**0.20**
SlideShow	0.18	**-0.03**	0.40	4.22	-0.06	0.51	3.42	-0.04	0.50	**0.05**	**-0.03**	**-0.04**
Average	1.25	-0.03	0.53	2.59	-0.07	0.89	2.39	-0.03	0.61	**0.84**	**-0.01**	**-0.03**
SD	0.79	0.03	0.70	1.75	0.05	0.98	1.85	0.04	0.92	**0.71**	**0.01**	**0.29**

TABLE VII

PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND STATE-OF-THE-ART ALGORITHMS WITH 60% EXPECTED COMPLEXITY

Sequences	Deng_ACCESS			Deng_TCSVT			Amaya_TMM			Proposed		
	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)
BasketballDrive	**0.24**	-0.15	2.02	2.12	-0.20	4.29	11.18	-0.17	3.59	1.71	**-0.04**	**0.25**
BQTerrace	1.90	-0.08	3.20	3.90	-0.19	7.33	5.33	-0.25	6.21	**0.37**	**-0.03**	**-0.94**
ParkScene	**1.27**	-0.14	5.87	4.78	-0.34	8.99	4.67	-0.31	7.22	2.17	**-0.04**	**0.07**
Johnny	0.90	-0.00	-0.02	2.23	-0.03	2.24	5.18	-0.00	-0.10	**0.54**	**-0.02**	**-0.63**
FourPeople	**0.23**	-0.01	0.44	3.09	-0.11	3.64	7.78	-0.03	0.68	1.34	**-0.03**	**-0.62**
Vidyo1	2.19	-0.02	0.62	5.22	-0.08	1.03	9.64	-0.02	0.24	**0.44**	**-0.02**	**-0.40**
Vidyo3	3.36	-0.05	1.55	5.31	-0.07	2.87	9.72	-0.03	0.66	**0.72**	**-0.02**	**0.19**
Vidyo4	1.01	**-0.01**	0.21	3.32	-0.04	1.36	9.99	-0.04	0.87	**0.36**	-0.02	**0.07**
PartyScene	**0.50**	-0.25	8.29	3.22	-0.53	14.10	2.95	-1.01	25.12	1.31	**-0.04**	**0.83**
RaceHorses	1.87	-0.45	7.37	**1.87**	-0.65	15.39	7.17	-0.53	14.20	2.16	**-0.03**	**0.71**
BasketballDrill	2.71	-0.18	4.85	3.06	-0.78	8.54	12.99	-0.24	5.19	**0.77**	**-0.03**	**0.54**
SlideEditing	1.06	-0.30	3.32	1.85	-0.56	6.79	4.11	-0.52	7.43	**0.38**	**-0.02**	**-0.06**
SlideShow	2.00	-0.06	0.58	2.40	-0.15	1.36	7.48	-0.11	0.42	**0.81**	**-0.05**	**0.33**
Average	1.48	-0.13	2.95	3.26	-0.29	5.99	7.55	-0.25	5.52	**1.01**	**-0.03**	**0.03**
SD	0.92	0.13	2.73	1.17	0.25	4.57	2.91	0.28	6.93	**0.64**	**0.01**	**0.52**

always the smallest, which indicates the proposed method outperforms these three state-of-the-art algorithms in terms of control accuracy. In fact, both Deng-TCSVT and Deng-ACCESS always encode a CTU from Depth 0 and control the encoding complexity by only adjusting the maximum CTU depth, which is inefficient to reasonably utilize the constrained complexity budgets. As for Amaya-TMM, its complexity control depends entirely on the feedback mechanism and this work only employs simple complexity estimation, which is based on the assumption that each CTU has the same cost for encoding complexity. By contrast, the proposed method can achieve the best control accuracy by using the following strategies: 1) multi-stage complexity allocation, which can reasonably allocate the expected complexity to each coding level; 2) CTU-level complexity control, which can achieve accurate complexity reduction; 3) the feedback mechanism

TABLE VIII

PERFORMANCE COMPARISON BETWEEN THE PROPOSED METHOD AND STATE-OF-THE-ART ALGORITHMS WITH 40% EXPECTED COMPLEXITY

Sequences	Deng_ACCESS			Deng_TCSVT			Amaya_TMM			Proposed		
	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)	CE (%)	ΔPSNR (dB)	ΔBR (%)
BasketballDrive	**0.37**	-0.28	6.03	4.32	-0.47	13.21	11.14	-0.85	15.73	1.23	**-0.10**	**2.82**
BQTerrace	1.62	-0.31	6.59	2.23	-0.65	14.87	6.99	-0.81	17.56	**0.95**	**-0.06**	**-0.33**
ParkScene	**0.02**	-0.34	8.24	1.39	-1.57	16.73	4.82	-0.94	15.20	0.66	**-0.08**	**1.16**
Johnny	1.74	-0.06	2.21	2.78	-0.12	4.54	4.78	-0.21	5.72	**0.65**	**-0.05**	**0.25**
FourPeople	1.10	-0.24	6.50	4.73	-0.44	10.53	6.81	-0.35	10.78	**0.43**	**-0.07**	**1.45**
Vidyo1	1.20	**-0.05**	1.89	2.68	-0.22	10.54	12.91	-0.23	9.89	**0.33**	-0.06	**0.71**
Vidyo3	1.63	-0.11	3.67	2.66	-0.23	11.31	8.07	-0.28	10.32	**1.64**	**-0.07**	**1.66**
Vidyo4	**0.02**	-0.10	3.77	1.79	-0.27	12.23	6.11	-0.21	5.87	1.69	**-0.06**	**1.32**
PartyScene	1.42	-1.03	13.24	5.20	-1.57	28.16	6.47	-1.65	33.21	**0.72**	**-0.08**	**3.79**
RaceHorses	2.45	-1.34	11.12	5.53	-1.77	27.49	4.61	-2.03	35.78	**1.54**	**-0.09**	**4.51**
BasketballDrill	2.72	-0.27	6.36	2.20	-0.68	13.27	4.53	-1.28	15.77	**1.27**	**-0.10**	**4.15**
SlideEditing	1.65	-0.87	6.54	6.06	-1.37	16.38	6.69	-0.97	14.82	**0.20**	**-0.04**	**0.32**
SlideShow	1.12	-0.12	**1.83**	5.00	-1.24	15.26	3.49	-0.78	12.99	**0.90**	**-0.22**	2.65
Average	1.31	-0.39	6.00	3.58	-0.82	14.96	6.72	-0.87	15.66	**0.94**	**-0.08**	**1.88**
SD	0.79	0.40	3.32	1.53	0.58	6.27	2.59	0.56	8.79	**0.48**	**0.04**	**1.51**

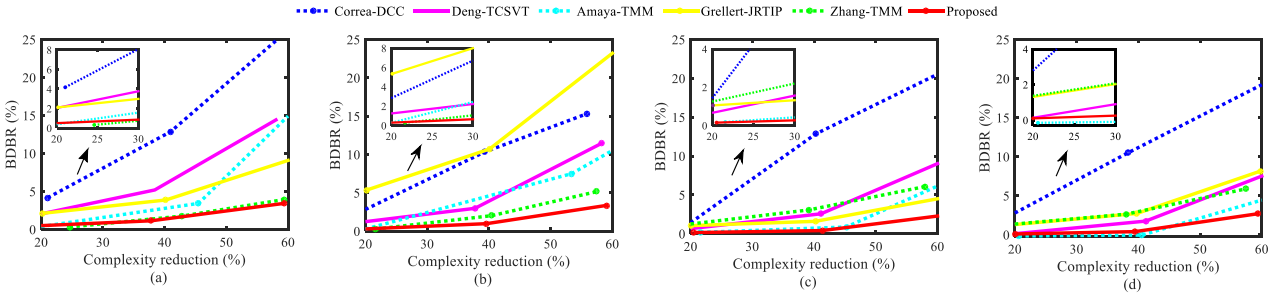


Fig. 12. BDBR versus encoding complexity reduction. (a) *BQTerrace* (b) *ParkScene* (c) *Johnny* (d) *FourPeople*.

which is utilized to eliminate the complexity control error of the previous CTUs.

F. Comparison of RD Performance

In Tables VI-VIII, the average values of ΔBR and ΔPSNR for the proposed method with three target complexity ratios are always the best, which indicates the proposed method is superior to the state-of-the-art algorithms in terms of RD performance. For Deng-TCSVT, the complexity control strategy relies entirely on the distortion, and some unreasonable early-skip results in RD performance loss. Although Deng-ACCESS introduces a bit map to guarantee the objective quality, the RD performance loss is still not negligible. The RD performance of Amaya-TMM degrades rapidly as the expected complexity ratio decreases, which is mainly because of the lack of an effective complexity allocation mechanism. By comparison, the reasonable complexity allocation strategy of the proposed method achieves the expected complexity by selecting the appropriate weight for the CU decision model. This approach maintains the trade-off between RD performance and complexity reduction efficiently and flexibly.

G. Evaluation of Complexity-Bitrate Performance

The increase of bitrate at varying complexity reduction was evaluated. Fig. 12 plots the complexity-bitrate curves of four sequences with different resolutions and video contents, for the proposed method and five other state-of-the-art methods (denoted as Correa-DCC, Deng-TCSVT, Amaya-TMM, Grellert-JRTIP, and Zhang-TMM, respectively). Among these five methods, Correa-DCC [23] is a classical complexity control method and Grellert-JRTIP [27] achieves complexity control by adjusting the encoding parameters. Zhang-TMM [29]] is a recent process simplification-based method. As shown in Fig. 12, the BDBR of the proposed method for different complexity reductions are always less than those of other methods, which indicates better compression efficiency.

H. Performance Evaluation With Various Length of Segment

The length of segment is determined by the number of GOPs (Z) in a segment. To investigate the influence of Z on the overall performance, we perform experiments on two sequences, *KristenAndSara* and *BQTerrace*. Fig. 13 shows the BDBR and CE variations of the proposed method with respect to various Z, i.e., 2, 4, 6, 8, and 10. On the one hand, Z slightly affects the BDBR. On the other hand, CE varies with different trend for different expected complexity when Z increases. When Z is quite small, the large number of updating frames will lead to high complexity overhead and more control error. In addition, when the proposed method is configured with a quite large Z, the updating frequency decrease. Scene variation will reduce the control accuracy. Therefore, it is necessary to selected appropriate length of

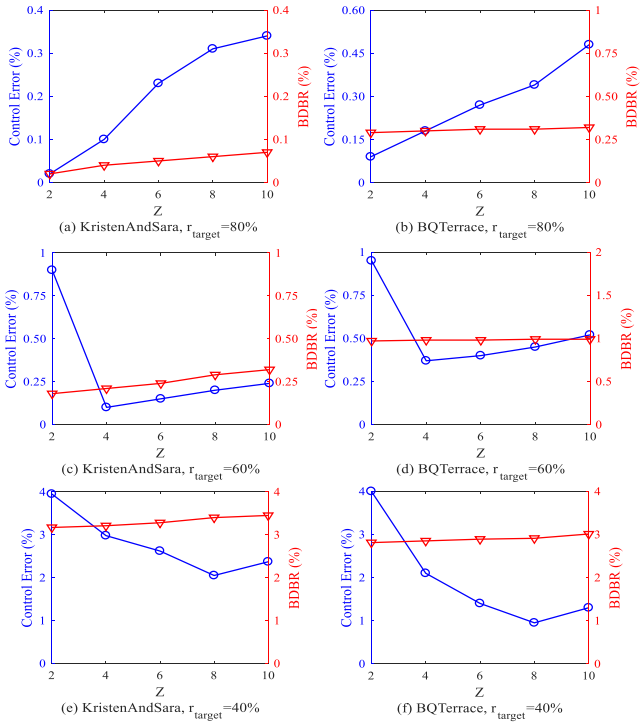


Fig. 13. Changes in control error and BDBR when the proposed method is configured with different number of GOPs in a segment.

segment for the proposed method with different expected complexity. We empirically set Z to 2, 4 and 8 for different expected complexity configurations. Specifically, Z are set to 2, 4, and 8 when the expected complexity ratio r_{Target} satisfies the conditions, $40\% \leq r_{Target} < 60\%$, $60\% \leq r_{Target} < 80\%$, and $80\% \leq r_{Target} \leq 100\%$, respectively.

VI. Conclusion

In this article, we propose an online learning-based multi-stage complexity control method to achieve a tunable trade-off between encoding complexity and RD performance for efficiently implementing HEVC on live video applications with different computing capacities and constrained power. The novelty of this approach lies in that the random forest-based CU decision model can adaptively address the problem of complexity control and the encoder can achieve good RD performance under the given constrained complexity. Moreover, the multi-stage complexity allocation strategy is well designed to reasonably allocate the encoding complexity to each coding level for obtaining a better control accuracy. The experimental results confirmed that the proposed method outperforms state-of-the-art approaches in terms of complexity control accuracy and RD performance. In future, we plan to apply the machine-learning-based multi-accuracy CU decision model into HEVC intra coding for industrial video applications.

References

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

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, 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. Correa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Performance and computational complexity assessment of high-efficiency video encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1899–1909, Dec. 2012.

[4] Q. Jiang, F. Shao, W. Lin, K. Gu, G. Jiang, and H. Sun, "Optimizing multistage discriminative dictionaries for blind image quality assessment," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 2035–2048, Aug. 2018.

[5] Q. Jiang, F. Shao, W. Lin, and G. Jiang, "Learning a referenceless stereopair quality engine with deep nonnegativity constrained sparse autoencoder," *Pattern Recognit.*, vol. 76, pp. 242–255, Apr. 2018.

[6] L. Shen, Z. Zhang, and Z. Liu, "Effective CU size decision for HEVC intracoding," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232–4241, Oct. 2014.

[7] 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.

[8] T. Mallikarachchi, D. S. Talagala, H. K. Arachchi, and A. Fernando, "Content-adaptive feature-based CU size prediction for fast low-delay video encoding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 693–705, Mar. 2018.

[9] Y. Zhang, S. Kwong, G. Zhang, Z. Pan, H. Yuan, and G. Jiang, "Low complexity HEVC INTRA coding for high-quality mobile video communication," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1492–1504, Dec. 2015.

[10] S. Ryu and J. Kang, "Machine learning-based fast angular prediction mode decision technique in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5525–5538, Nov. 2018.

[11] C. Huang, Z. Peng, F. Chen, Q. Jiang, G. Jiang, and Q. Hu, "Efficient cu and pu decision based on neural network and gray level co-occurrence matrix for intra prediction of screen content coding," *IEEE Access*, vol. 6, pp. 46643–46655, 2018.

[12] 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.

[13] C. C. Wang, Y. C. Liao, J. W. Wang, and C. W. Tung, "An effective TU size decision method for fast HEVC encoders," in *Proc. Int. Symp. Comput., Consum. Control*, Jun. 2014, pp. 1195–1198.

[14] 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.

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

[16] K. Miyazawa, T. Murakami, A. Minezawa, and H. Sakate, "Complexity reduction of in-loop filtering for compressed image restoration in HEVC," in *Proc. Picture Coding Symp.*, May 2012, pp. 413–416.

[17] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective data driven coding unit size decision approaches for HEVC INTRA coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3208–3222, Nov. 2018.

[18] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: A deep learning approach," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5044–5059, Oct. 2018.

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

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

[21] X. Liu, Y. Li, D. Liu, P. Wang, and L. T. Yang, "An adaptive CU size decision algorithm for HEVC intra prediction based on complexity classification using machine learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 144–155, Jan. 2019.

[22] G. Correa, P. Assuncao, L. Agostini, and L. da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.

[23] G. Correa, P. Assuncao, L. Agostini, and L. A. D. S. Cruz, "Coding tree depth estimation for complexity reduction of HEVC," in *Proc. Data Compress. Conf.*, Mar. 2013, pp. 43–52.

[24] 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.
[25] X. Deng, M. Xu, and C. Li, "Hierarchical complexity control of HEVC for live video encoding," *IEEE Access*, vol. 4, pp. 7014–7027, 2016.
[26] A. Jimenez-Moreno, E. Martinez-Enriquez, and F. Diaz-de-Maria, "Complexity control based on a fast coding unit decision method in the HEVC video coding standard," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 563–575, Apr. 2016.
[27] M. Grellert, B. Zatt, M. Shafique, S. Bampi, and J. Henkel, "Complexity control of HEVC encoders targeting real-time constraints," *J. Real-Time Image Process.*, vol. 13, no. 1, pp. 5–24, Mar. 2017.
[28] G. Correa, P. A. Assuncao, 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.
[29] J. Zhang, S. Kwong, T. Zhao, and Z. Pan, "CTU-level complexity control for high efficiency video coding," *IEEE Trans. Multimedia*, vol. 20, no. 1, pp. 29–44, Jan. 2018.
[30] S. Zhu and D. Zhao, "Fast intra-prediction mode decision algorithm for high efficieny video coding," in *Proc. 9th IEEE Conf. Ind. Electron. Appl.*, Jun. 2014, pp. 936–939.
[31] Y. Gao, C. Zhu, S. Li, and T. Yang, "Temporally dependent rate-distortion optimization for low-delay hierarchical video coding," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4457–4470, Sep. 2017.
[32] N. Quadrianto and Z. Ghahramani, "A very simple safe-Bayesian random forest," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1297–1303, Jun. 2015.
[33] L. Breiman, J. Friedman, C. Stone, and R. Olshen, "Classification and regression trees," *Encyclopedia Ecol.*, vol. 57, no. 3, pp. 582–588, 2015.
[34] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEGM33, ITU-T SG 16/Q6, 13th VCEG Meeting, Austin, TX, USA, Apr. 2001.

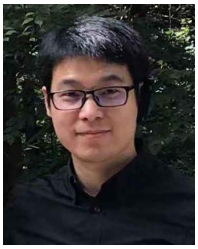
Chao Huang received the B.S. and M.S. degrees from Ningbo University in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. His research interests include image/video coding, video analysis, visual anomaly detection, and deep learning.

Zongju Peng (Member, IEEE) received the B.S. degree from the Sichuan Normal College, China, in 1995, the M.S. degree from Sichuan University, China, in 1998, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2010. He is currently a Professor with the School of Electrical and Electronic Engineering, Chongqing University of Technology, Chongqing, China. His main research interests include image/video compression, multi-view video coding, and video perception.

Yong Xu (Senior Member, IEEE) received the B.S. and M.S. degrees in 1994 and 1997, respectively, and the Ph.D. degree in pattern recognition and intelligence system from NUST, China, in 2005. He currently works with the Harbin Institute of Technology, Shenzhen. His current research interests include pattern recognition, biometrics, machine learning, and video analysis. More information please refer to http://www.yongxu.org/lunwen.html

Fen Chen received the B.S. degree from the Sichuan Normal College, China, in 1996, and the M.S. degree from the Institute of Optics and Electronics, Chinese Academy of Sciences, in 1999. She is currently a Professor with the School of Electrical and Electronic Engineering, Chongqing University of Technology, Chongqing, China. Her main research interests include digital signal processing, image processing and communications, and multi-view video processing.

Qiuping Jiang (Member, IEEE) received the Ph.D. degree in signal and information processing from Ningbo University, Ningbo, China, in 2018. From 2017 to 2018, he was a Visiting Student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor with the School of Information Science and Engineering, Ningbo University. His research interests include perceptual-driven image processing, computer vision, and machine learning.



Yun Zhang (Senior Member, IEEE) 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 Postdoctoral Researcher with the Department of Computer Science, City University of Hong Kong, Hong Kong. From 2010 to 2017, he was an Assistant Professor and an Associate Professor with the Shenzhen Institutes of Advanced Technology (SIAT), CAS, Shenzhen, China, where he is currently a Professor. His research interests include video compression, 3D video processing, and visual perception.

Gangyi Jiang (Senior Member, IEEE) received the M.S. degree from Hangzhou University in 1992 and the Ph.D. degree from Ajou University, South Korea, in 2000. He is currently a Professor with the Faculty of Information Science and Engineering, Ningbo University, China. His main research interests include digital video compression and communications, multi-view video coding, and image processing.

Yo-Sung Ho (Fellow, IEEE) received the B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, South Korea, in 1981 and 1983, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1990. Since 1995, he has been with the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. Since 2003, he has been the Director of the Realistic Broadcasting Research Center, GIST, where he is currently a Professor with the School of Electrical Engineering and Computer Science. His research interests include digital image and video coding, 3D image modeling and representation, augmented reality and virtual reality, and realistic broadcasting technologies. He has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the IEEE TRANSACTIONS ON MULTIMEDIA.