Low Complexity HEVC INTRA Coding for High-Quality Mobile Video Communication

Yun Zhang, *Member, IEEE*, Sam Kwong, *Fellow, IEEE*, Guangjun Zhang, Zhaoqing Pan, *Member, IEEE*, Hui Yuan, *Member, IEEE*, and Gangyi Jiang, *Member, IEEE*

***Abstract*—INTRA video coding is essential for high quality mobile video communication and industrial video applications since it enhances video quality, prevents error propagation, and facilitates random access. The latest high-efficiency video coding (HEVC) standard has adopted flexible quad-tree-based block structure and complex angular INTRA prediction to improve the coding efficiency. However, these technologies increase the coding complexity significantly, which consumes large hardware resources, computing time and power cost, and is an obstacle for real-time video applications. To reduce the coding complexity and save power cost, we propose a fast INTRA coding unit (CU) depth decision method based on statistical modeling and correlation analyses. First, we analyze the spatial CU depth correlation with different textures and present effective strategies to predict the most probable depth range based on the spatial correlation among CUs. Since the spatial correlation may fail for image boundary and transitional areas between textural and smooth areas, we then present a statistical model-based CU decision approach in which adaptive early termination thresholds are determined and updated based on the rate-distortion (RD) cost distribution, video content, and quantization parameters (QPs). Experimental results show that the proposed method can reduce the complexity by about 56.76% and 55.61% on average for various sequences and configurations; meanwhile, the RD degradation is negligible.**

***Index Terms*—Coding unit (CU), high-efficiency video coding (HEVC), low complexity, power efficient, spatial correlation.**

Manuscript received January 19, 2015; revised August 13, 2015; accepted September 28, 2015. Date of publication October 15, 2015; date of current version December 02, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61471348, Grant 61272289, Grant 61571274, and Grant U1301257; in part by the Shenzhen Overseas High-Caliber Personnel Innovation and Entrepreneurship Project under Grant KQCX20140520154115027; and in part by the Guangdong Special Support Program for Youth Science and Technology Innovation Talents under Grant 2014TQ01X345. Paper no. TII-15-0082.

Y. Zhang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: yun.zhang@siat.ac.cn).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong, and also with the Shenzhen Research Institute, City University of Hong Kong, Shenzhen 518057, China (e-mail: cssamk@cityu.edu.hk).

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

Z. Pan is with the Jiangsu Engineering Center of Network Monitoring, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: zhaoqingpan@nuist.edu.cn).

H. Yuan is with the School of Information Science and Engineering, Shandong University, Ji'nan 250100, China (e-mail: yuanhui0325@gmail.com).

G. Jiang is with the Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China (e-mail: jianggangyi@nbu.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/TII.2015.2491646

I. Introduction

With the development of multimedia technologies and pervasive computing technologies, the past few decades have witnessed a great success of the development of various mobile devices and applications, which significantly facilitates people's life and industrial manufacturing. As mobile devices, such as smart phones, personal digital assistant (PDA), and tablet, getting more and more popular, it booms broadband mobile Internet access, e.g., 3G and 4G, and thus arises the demand for mobile video applications, such as mobile video communications, remote monitoring, visual sensor network, surveillance, screen sharing, recording, editing, mobile 3-D, and mobile TV. Owning to an increasing demand for high visual quality, high definition (HD) and ultra HD (UHD) videos become popular and are inevitable new trends of consumer and industrial video applications since they can provide more realistic visual enjoyment and more accurate representation beyond human eyes. However, the HD/UHD video data volume increases dramatically as the increase of video resolution (up to $4k \times 2k,\ 8k \times 4k$) and frame rate (e.g., 60 fps and up to 600 fps for industrial high-speed video), which requires a powerful encoder to compress them much more efficiently in order to reduce the service charge, data traffic, and meanwhile enhance the service quality.

Besides the coding efficiency, video error resilience as well as coding complexity shall also be considered in encoder design [1], [2]. In the mobile video and related industrial applications, network latency, capabilities, and bandwidth are not stable and usually fluctuate due to various environments, capacity of devices, and heterogeneous network conditions [3], [4], etc. Moreover, the wireless network for video transmission is an error prone channel where the packet loss as well as bit error may easily occur. These error data not only degrade the image quality of the error/lost frame but also bring about distortions to successive frames due to INTER prediction and motion compensation, which is called error propagation. However, these transmission error cannot be recovered by retransmissions or nondeterministic back-offs due to real-time requirements. To enhance the video quality and prevent the error propagation, INTRA or refresh INTRA frames, which do not reference to other previous coded frames, are added while encoding. On the other hand, the INTRA frames are random access points that allow a decoder to start decoding properly at the location of INTRA frames, which is useful in video skimming. Another advantage of using INTRA frame is that it has much lower complexity than using the INTER frames, i.e., P and B frames, which is very important to machine vision and industrial high speed video that up to 600 fps. This property is essential to mobile/portable devices with limited resources, such as computing capabilities, memory, and battery.

An high-efficiency video coding (HEVC)/H.265 [5] has been standardized as the latest video coding standard, which doubles the compression efficiency of H.264/MPEG-4 AVC high profile [6], i.e., half the transmission bit rate or storage space while maintaining the same video quality. While improving the compression efficiency of INTRA frames, many new coding technologies have been adopted, including sophisticated INTRA prediction modes (up to 35 modes in HEVC), multi-mode transform, and hierarchical coding block structure [5]. However, they significantly increase the computational complexity and memory access of the encoder, which consumes more computing time, power, and battery of the mobile devices. Meanwhile, larger resolution and higher frame rate of HD/UHD video further increase the data volume for compressing, which makes power efficient video coding more challenging and desirable. In a word, a low power and high efficiency INTRA coding is essentially important and highly desired to mobile videos and industrial video applications, such as remote monitoring, surveillance, remote control, machine vision, and industrial video camera with high-speed imaging.

II. Related Works

With the development of video coding technologies, dozens of mode candidates are included in order to adapt to diverse video contents and maximize the coding efficiency [5], [6]. The optimal mode among the candidates will be selected to achieve the best performance. Mathematically, the optimal mode m^* among the candidate set $\mathbf{M}$ is obtained by minimizing the Lagrangian cost J, which is presented as

$$\begin{cases} m^* = \arg\min_{m\in\mathbf{M}} J(m) \\ J(m) = D(m) + \lambda R(m) \end{cases} \tag{1}$$

where $D(m)$ is the distortion between the original block and the reconstructed block while using coding mode m, $R(m)$ is encoding bits with m, and λ is the Lagrange multiplier. In fact, the determination of the modes is a decision problem, which is common and prevalent in coding technologies, including block partitioning, INTRA and INTER prediction, reference frame selection, transform, filtering, and motion estimation. In the original test model, the encoder tries all candidates in $\mathbf{M}$ and calculates the cost of each mode, i.e., $J(m)$, then, the mode m who generates the minimum cost is selected as the optimal. This "try all and select the best" strategy is able to find the optimal but is time-consuming since all candidates shall be checked.

By analyzing the developing trend of the standards from MPEG-1, MPEG-2, H.264/AVC to HEVC, more and more refined mode candidates or parameters have been adopted in encoding modules. For example, the number of INTRA prediction mode is 0 in MPEG-1/2 and 9 in H.264/AVC [6], and it increases to 35 in HEVC [5]. The number of block type is 1 in MPEG-1 and 7 in H.264/AVC [6]. In HEVC, there are 85 leaf nodes in the quad-tree of the coding unit (CU) size decision [5], [7], which means the encoder shall calculate $J(m)$ 85 times as finding the optimal CU depth. Moreover, there are 11 prediction unit (PU) modes in each CU, including nine INTER modes and two INTRA modes. Then, the transform module tries three transform unit (TU) sizes (32×32, 16×16, and 8×8) in each PU. There are three recursive loops in the mode decision in the HEVC encoder, i.e., CU, PU, and TU. The number of mode candidates increases significantly compared to the previous standards. Thus, the complexity, memory access, and power cost increase dramatically.

Many researchers have devoted their efforts on reducing the HEVC encoding complexity [2], [7]–[14]. To determine the optimal CU partition more efficiently, the CU depth range was predicted from those of spatial neighboring CUs and temporal colocated CU [7]. Then, the motion characteristic and rate-distortion (RD) cost of the temporal corresponding block were used to refine the CU depth prediction. Lei *et al.* [2] exploited inter-view [1] and intercomponent correlations in fast CU size decision for HEVC-based 3-D depth map coding. As the CU decision can be modeled as classification problems, learning algorithms [8]–[14] were investigated and applied to the low complexity HEVC optimization. Xiong *et al.* [8] determined the optimal CU based on unsupervised K-nearest clustering via pyramid motion divergence. Peixoto *et al.* [9] proposed a fast CU depth decision in transcoding from H.264/AVC to HEVC, where linear discriminant functions (LDFs) were applied and bit stream was extracted as additional features. In [10], Shen and Yu modeled the quad-tree CU depth decision as several binary classifications and early terminated the CU decision process with weighted support vector machine (SVM). Besides, Bayesian decision rules [11], Markov random field [12] and decision tree [13] models were also used in HEVC CU size decision. Zhang *et al.* [14] proposed a machine learning-based fast CU depth decision framework, which included hierarchical decision structures, three-output joint SVM classifier, and the optimal learning parameter determination to minimize the coding complexity with given RD cost constraints. They basically belong to CU early termination algorithms for fast HEVC INTER coding.

In addition, Pan *et al.* [15] early determined the merge mode based on motion and hierarchical CU correlation. Shen *et al.* [16] proposed a fast INTER PU mode decision scheme by jointly using interdepth correlation and spatiotemporal correlation in terms of mode, motion vector, and the RD cost. Vanne *et al.* [17] analyzed the RD-complexity impacts of HEVC INTER prediction techniques, and skipped some of the symmetric motion partition (SMP) and asymmetric motion partition (AMP) mode at PU level. Lee *et al.* [18] determined whether skip PU modes by utilizing the RD cost of $2N \times 2N$ merge mode. These schemes were proposed for fast INTER PU mode optimization.

Since the INTRA coding is also highly complex and has less available information, several works [19]–[24] have been done for fast INTRA CU decision. In [19], the INTRA CU depth range was adapted at slice/CU level based on the CU depth of previous coded slices/CUs. In [20], a fast INTRA CU depth decision was proposed based on the texture homogeneity as well as spatial neighboring coded CUs. Wang *et al.* [21] determined the CU size by jointly exploring correlations between neighboring coding tree units (CTUs), the RD cost, and sum of absolute difference (SAD) with Hadamard transform. Then, rough mode decision (RMD) was improved in each CU to reduce the angular mode prediction complexity. However, the spatial correlation may fail in image boundaries or transitional areas. Therefore, in [22], the video texture characteristics and coding bits of each CU were adopted to facilitate the CU depth decision. In addition, the CU size is highly correlated with the RD cost. Cho and Kim [23] presented an early CU splitting and pruning decision for INTRA coding. The splitting and pruning tests were performed at each CU depth level by using a Bayes decision rule method, whose statistical parameters were periodically updated on the fly to cope with varying signal characteristics. In [24], a fast joint CU and angular mode prediction was proposed for INTRA coding by using the difference between the minimum and the second minimum RD cost estimations from Hadamard transform. These schemes are CU size decision for INTRA coding and still some room for further improvement on feature exploration and the coding optimization.

In addition to the INTRA CU size decision, some works on PU and angular mode decision from up to 35 modes have also been investigated in [25]–[30] for fast INTRA coding. Lim *et al.* [25] proposed fast PU skip and split termination algorithm, which includes early skip, PU skip, and PU split termination, by considering neighboring PUs and RMD cost differences. Zhang and Zhan [26] proposed Hadamard cost-based progressive RMD and early RD optimized quantization skip method to facilitate the angular INTRA prediction. Hu and Yang [27] jointly optimized CU size and mode decision by using transparent composite model of discrete cosine transform (DCT) coefficients and outlier information of the model. To effectively predict the optimal INTRA mode from 35 mode candidates, the variance of the neighboring reference samples [28], edge and sum of absolute transform difference (SATD) information [29] were exploited to reduce the number of searching candidates. In [30], edges identified by Hadamard transform were considered in fast INTRA depth coding for 3-D video system.

In this work, we propose an advanced INTRA CU decision algorithm to effectively lower the computational complexity of HEVC by jointly considering the spatial correlation, video texture, and statistical RD cost properties. The novelty of the proposed algorithm lies in the following two aspects: 1) subalgorithm spatial correlation-based CU depth decision (SC-CUDD) jointly exploits the CU's texture and spatial correlation, in which the hit rate (HR) and complexity reduction of 7–9 prediction strategies are analyzed in detail to obtain the optimal one. 2) Subalgorithm statistical model-based CU early termination (SM-CUET) is proposed by exploiting the statistical RD properties of the CUs, in which early termination thresholds can be adaptively determined and updated based on the RD cost distribution, video contents, and quantization parameters (QPs). The proposed two schemes are mutually complementary and combined to effectively reduce the coding complexity, which can promote the HD video codec design and the related industrial applications. This paper is organized as follows. Motivation and upper-bound of complexity reduction are presented in Section III. Then, fast INTRA CU depth decision algorithms consisting of SC-CUDD and SM-CUET are presented in Section IV. Experimental results and analyses are presented in Section V. Finally, conclusion is drawn in Section VI.

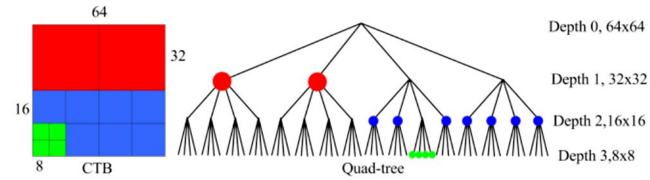


Fig. 1. Quad-tree coding structure of CU partition.

III. Motivations and Complexity Analysis

As shown in Fig. 1, there are $1 + 4 + 16 + 64 = 85$ leaf nodes in the quad-tree while the tree depth is from depth 0 to depth 3. The size of red, blue, and green CUs is 32×32, 16×16, and 8×8, respectively. In HEVC, only some of the nodes in the tree will be selected as the optimal CU partition for a given coding tree block (CTB). The color dots are in correspondence with the left CTB partitions, and $2 + 7 + 4 = 13$ leaf nodes are finally selected. In this example, 85 leaf nodes are checked but only 13 of them are finally selected. The best case, i.e., the minimum number of nodes, is only one node will be selected, where the best CU size is 64×64 and the best depth is 0. The worst case, i.e., maximum number of nodes, is 64 nodes will be selected, in which the best CU depth is 3. It means only 1–64 nodes will be finally selected as the optimal CU partition. However, we shall check and calculate the RD cost of all 85 nodes for each CTB in the original HEVC, which has many unnecessary operations. In other words, if we can efficiently predict the CU depth/size, significant complexity reduction can be achieved. The theoretical upper-bound of the computational complexity reduction (CCR) $\Delta C_{\max}$ for each CTB can be obtained under the assumption that the CU size is precisely predicted, which is mathematically presented as

$$\Delta C_{\max} = 1 - \frac{\sum_{k=1}^{N_k} \sum_{i=0}^{j} n_i(k) \cdot C_i}{N_k \sum_{i=0}^{j} 4^i \cdot C_i} \tag{2}$$

where i is the depth of CU, $n_i(k)$ is the number of the nodes at depth i selected as the optimal CU for one CTB k, suppose the coding complexity of the nodes at depth i is same and denoted by C_i, j is the maximum allowed CU depth, and N_k is the total number of CTBs.

To evaluate the complexity redundancies in HEVC, we statistically analyze different video sequences coded by the HEVC and calculate the upper-bound of their CCRs. Five different sequences were encoded by the HEVC encoder with four QPs, which are 22, 27, 32, and 37. 100 frames for each sequences are encoded by HM8.0 [31] with all INTRA main (AIM) profile. The rest settings follow the common test conditions (CTCs) [32]. Fig. 2 shows the average CU depth distribution over different QPs for five typical test sequences, we can observe that

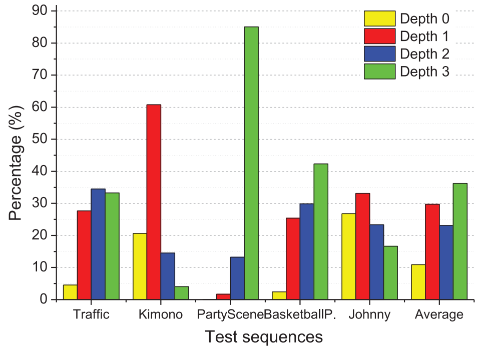


Fig. 2. CU depth distribution for different sequences.

TABLE I
MAXIMUM POTENTIAL CCR $\Delta C_{\max}$ FOR DIFFERENT QPS AND SEQUENCES (UNIT: %)

Test sequences	QP22	QP27	QP32	QP37	Average
Traffic	64.38	68.81	72.60	77.24	70.76
Kimono	83.69	85.43	86.44	87.24	85.70
PartyScene	45.66	46.39	48.45	52.24	48.19
BasketballPass	60.51	63.99	67.21	71.61	65.83
Johnny	76.40	78.45	80.01	82.10	79.24
Average	**66.13**	**68.61**	**70.94**	**74.09**	**69.94**

the depth 1 dominates in the *Kimono*, depth 3 dominates in the *PartyScene*, and some are evenly distributed. Generally, the CU depth distributions vary with contents and correlations, which makes CU depth prediction very challenging. Suppose that the CU size of the video sequence is 100% precisely predicted, the $\Delta C_{\max}$ for different sequences are shown in Table I. We have the following three observations.

1) The potential CCR increases as the QP increases since more CTBs will select larger sizes as the optimal CU size.
2) The $\Delta C_{\max}$ varies as the video content changes. Usually, the potential CCR decreases as the texture of video content becomes complex.
3) The average $\Delta C_{\max}$ over different QPs and sequences is 69.94%, which indicates there are a large amount of complexity redundancies in HEVC INTRA coding and they shall be removed.

Since the original mode decision process in HEVC is "try all and select the best" and extremely redundant, our objective for optimizing (1) is to develop a fast mode decision algorithm that maximizes CCR and make it approaches $\Delta C_{\max}$ subject to negligible RD degradation (ΔJ_T), which can be presented as

$$\max \text{CCR}, \text{ subject to } J(m^+) - J(m^*) < \Delta J_T \tag{3}$$

where CCR increases as the number of searched candidates reduces, m^+ is the suboptimal or optimal mode by using a fast decision algorithm. $J(m^+)$ satisfies $J(m^+) \geq J(m^*)$ since m^* is optimal and not worse than m^+.

IV. PROPOSED FAST CU DECISION BASED ON CORRELATION ANALYSES AND STATISTICS

In this section, two algorithms, correlation analyses and statistical model-based fast CU depth decision, are presented for INTRA coding. Then, the overall algorithm is presented.

TABLE II
ACCURACY OF PREDICTING $D(B_{\text{cur}})$ WITH $D(B_{\text{up}})$ OR $D(B_{\text{left}})$ (UNIT: %)

Prediction mode	Sequence	QP22	QP27	QP32	QP37	Average
$P(D(B_{\text{cur}})=D(B_{\text{left}}))$	Traffic	88.61	85.28	82.60	77.71	83.55
	Kimono	69.98	66.54	62.72	59.11	64.59
	PartyScene	97.03	96.88	96.61	95.56	96.52
	BasketballP.	91.37	86.83	77.59	76.40	83.05
	Johnny	80.23	73.47	80.68	80.81	78.80
	Average					**81.30**
$P(D(B_{\text{cur}})=D(B_{\text{up}}))$	Traffic	89.43	86.23	83.34	78.82	84.46
	Kimono	69.77	68.26	65.09	61.19	66.08
	PartyScene	97.73	97.10	96.62	95.81	96.82
	BasketballP.	80.51	75.77	64.27	61.52	70.52
	Johnny	76.94	69.02	73.86	80.81	75.16
	Average					**78.61**

A. Spatial Correlation-Based CU Depth Decision

Since the video content is highly spatially correlated, the CU depths of the spatial neighboring CTBs are highly correlated, which could be exploited for the CU decision. Usually, the spatial correlation decreases as the distance increases, the CU depth range of the above and left CTBs are more correlated to that of the current CTB. To analyze this correlation, we let B_{up} and B_{left} be the above and left CTBs, respectively; B_{cur} denotes the current CTB. Operator $D(X)$ denotes the depth range of CTB X. We analyze the HR of using $D(B_{\text{up}})$ and $D(B_{\text{left}})$ to predict the $D(B_{\text{cur}})$ over different test sequences and QPs, as shown in Table II. The experimental settings are the same as those of the previous statistical experiments in Section III. For example, $P(D(B_{\text{cur}}) = D(B_{\text{up}}))$ indicates the prediction accuracy of predicting $D(B_{\text{cur}})$ using $D(B_{\text{up}})$. We can observe that the prediction accuracies from B_{up} and B_{left} are 81.30% and 78.61% on average, respectively, for different types of sequences and QPs. This kind of direct predictions is not accurate enough to predict the CU depth range for current CTB; otherwise, large RD degradation will be caused in INTRA coding. Therefore, we need to develop new prediction strategies to improve the prediction accuracy.

As we know from the above analyses, the direct prediction from B_{up} or B_{left} is not so efficient. To tackle this problem, we jointly take the texture and spatial correlation into consideration since the INTRA CU depth is usually texture dependent [20]. For example, if the B_{cur} has more similar texture with the B_{up} when compared with B_{left}, it is likely that $D(B_{\text{cur}})$ is similar with $D(B_{\text{up}})$. On the other hand, if the texture of the B_{cur} is smoother or more complex than the textures of B_{up} and B_{left}, it shall be treated differently. Motivated by the texture properties of the video content, we divided a frame into three kinds of regions and proposed corresponding prediction strategies. The three regions are as follows.

1) Normal region is the B_{cur} whose texture complexity is in-between those of B_{up} and B_{left}, i.e., $\min(T(B_{\text{up}}), T(B_{\text{left}})) \leq T(B_{\text{cur}}) \leq \max(T(B_{\text{up}}), T(B_{\text{left}}))$.
2) Smooth region is the B_{cur} whose texture complexity is smoother than those of B_{up} and B_{left}, i.e., $T(B_{\text{cur}}) < \min(T(B_{\text{up}}),\ T(B_{\text{left}}))$.

TABLE III
PREDICTION STRATEGIES FOR THE CTB IN NORMAL REGION (UNIT: %)

Short Name	Prediction Strategies	Avg. HR	Avg. CCR
SN1	$D(B_{cur})=D(B_{up})\cap D(B_{left})$	67.54	19.43
SN2*	$D(B_{cur})=D(B_{up})\cup D(B_{left})$	94.20	10.20
SN3	$D(B_{cur})=D(B_{up})$	81.30	13.09
SN4	$D(B_{cur})=D(B_{left})$	78.61	16.67
SN5	$D(B_{cur})=D(B_{up})$ if $T(B_{up})>T(B_{left})$, otherwise, $D(B_{cur})=D(B_{left})$	82.01	15.09
SN6	$D(B_{cur})=D(B_{up})$ if $T(B_{up})<T(B_{left})$, otherwise, $D(B_{cur})=D(B_{left})$	78.35	14.67
SN7	$D(B_{cur})=D(B_{up})$, if $\lvert T(B_{cur})-T(B_{up})\rvert<\lvert T(B_{cur})-T(B_{left})\rvert$, otherwise, $D(B_{cur})=D(B_{left})$	81.89	15.41

3) Complex region is B_{cur} whose texture is more complex than those of B_{up} and B_{left}, i.e., $T(B_{\text{cur}}) > \max(T(B_{\text{up}}), T(B_{\text{left}}))$.

The operator $T(X)$ indicates the texture complexity of CTB X, which is calculated by

$$T(X) = \sum_{I_{ij}\in X} \left| I_{ij} - \frac{1}{N_X} \sum_{I_{ij}\in X} I_{ij} \right| \tag{4}$$

where N_X is the number of pixels in the CTB X, and I_{ij} is the luminance value of the pixel at position (i, j). The three kinds of regions will be analyzed individually and the optimal CU depth prediction strategies will be presented correspondingly.

1) Normal Region: We have developed and analyzed seven different CU depth prediction strategies (SN1 to SN7) for the CTB in this normal region, as shown in Table III. The union operator is the sum of the depth ranges of two blocks and intersection operator describes the common part of the two depth ranges. SN5 to SN7 predict the current CU depth by considering the texture similarity with neighboring CUs. For example, SN7 is $D(B_{\text{cur}})$ that is predicted from $D(B_{\text{up}})$ if the complexity difference between B_{cur} and B_{up} is smaller than that of B_{cur} and B_{left}. In order to evaluate the performance of these strategy candidates and select the best one, we implemented them on HEVC HM model and encoded five different sequences (*Traffic*, *Kimono*, *Johnny*, *PartyScene*, and *BasketballPass*) with AIM configuration, 100 frames for each sequence. Four basis QPs, QP $\in$ 22, 27, 32, 37, were tested. Two indices, the HR and CCR of each prediction strategy, were evaluated. Note that the CTB in complex and smooth regions is not optimized and counted in HR/CCR statistics. In addition, the HR indicates the ratio of correctly predicted CUs to total number of CUs in the normal region. The third and fourth columns of Table III show the average HR and CCR over different sequences and QPs. It is observed that the average HR varies from 67.54% to 94.20% for different strategies, and the average CCR is from 10.20% to 19.43%. The SN2 is with the highest HR, which is up to 94.20%, and it is about 12%–15% higher than those of SN3 and SN4.

Fig. 3 shows the detailed HR, CCR for different sequences. In the figure, each symbol indicates a strategy and each point indicates the performances of a sequence. We can find that SN2 is the best one in terms of the HR, even the lowest one (*Kimono*) is higher than 85%. Though the CCR achieved by SN2 is relatively lower than others, we still adopt it as the optimal one by giving higher priority to the RD performance (i.e., HR).

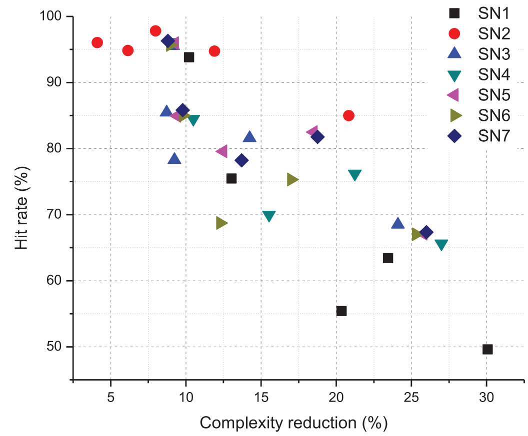


Fig. 3. HR and CCR for normal region.

TABLE IV
PREDICTION STRATEGIES FOR CTB IN COMPLEX AND SMOOTH REGIONS (UNIT: %)

Prediction Strategies	Complex region			Smooth region		
	Short Name	Avg. HR	Avg. CCR	Short Name	Avg. HR	Avg. CCR
$[\max(a_0-1,0),\max(a_1-1,0)]$	SC1	40.76	32.81	SS1	60.53	35.51
$[\max(a_0-1,0),a_1]$	SC2	96.51	23.38	SS2*	98.57	22.88
$[\max(a_0-1,0),\min(a_1+1,3)]$	SC3	99.50	21.64	SS3	99.82	20.86
$[a_0,\max(a_1-1,0)]$	SC4	38.19	34.80	SS4	53.34	38.01
$[a_0,a_1]$	SC5	93.94	25.38	SS5	91.06	25.38
$[a_0,\min(a_1+1,3)]$	SC6*	97.24	23.62	SS6	92.64	23.33
$[\min(a_0+1,3),\max(a_1-1,0)]$	SC7	19.78	37.28	SS7	18.23	41.07
$[\min(a_0+1,3),a_1]$	SC8	71.85	28.03	SS8	52.52	28.72
$[\min(a_0+1,3),\min(a_1+1,3)]$	SC9	74.84	26.27	SS9	54.10	26.66

2) Complex and Smooth Regions: For the complex and smooth regions, we developed nine strategies for the CTB depth range prediction, as shown in the first column in Table IV. The first column shows the nine different prediction strategies for the $D(B_{\text{cur}})$, where a_0 and a_1 are the minimum and maximum CU depth of $D(B_{\text{up}})$ and $D(B_{\text{left}})$, $a_0 \le a_1$. These prediction strategies are shortened as SC1 to SC9 for complex region and SS1 to SS9 for smooth region. For example, the strategy SC5 is $[a_0, a_1]$, which means the predicted depth range is from a_0 to a_1 for the complex regions. SC4 is $[a_0, \max(a_1 - 1, 0)]$ which indicates predicted depth range that is from a_0 to $a_1 - 1$. Since $a_0 \pm 1$ and $a_1 \pm 1$ might be out of the valid depth range [0,3], *max*() and *min*() are operators that clipped them to [0,3]. The rest strategies SCn for the complex region and strategies SSn, $n \in$[1,9], for the smooth region can be interpreted similarly. Similar to the process and settings of evaluating the prediction strategies in the normal region, we analyzed the prediction strategies SCn and SSn in terms of HR and CCR. As shown in Table IV, SC3 and SS3 are with the highest HR, which are 99.50% and 99.82%; however, these strategies are lowest in CCR. Fig. 4 shows HR and CCR for different sequences, where Fig. 4(a) and (b) is for complex and smooth regions, respectively. Basically, with respect to the average HR, SC2, SC3,

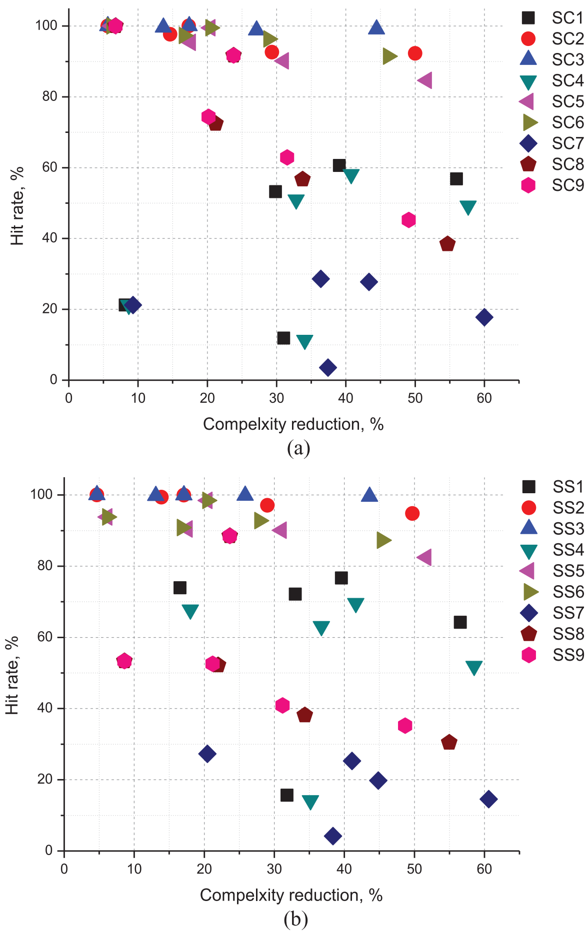


Fig. 4. HR and CCR for complex and smooth regions. (a) Complex region. (b) Smooth region.

SC5, and SC6 are also good for the complex region; SS2, SS3, SS5, and SS6 are good for smooth region. For the complex regions, the current CTB has higher probability of selecting smaller CU (larger CU depth) as its best CU comparing to its neighboring B_{up} and B_{left} CUs. Therefore, we shall further check a higher CU depth level, i.e., $a_1 + 1$. For the smooth regions, the current CTB is likely to select larger CU (smaller CU depth). Thus, we shall further check a lower CU depth level, i.e., $a_0 - 1$. In addition, SC6 and SS2 are both higher than 91.5% even with the worst case (*Kimono*). Therefore, to have a better tradeoff between the HR and CCR, we selected SC6 and SS2 as the optimal strategies for the complex and smooth regions in this paper, respectively.

Based on the above analyses, we derive the optimal CU size prediction strategy based on the spatial correlation and video texture as

$$D\left(B_{\text{cur}}\right) = \begin{cases} [\max(0, a_0 - 1), a_1], & T(B_{\text{cur}}) < \min(T(B_{\text{up}}), T(B_{\text{left}})) \\ [a_0, a_1], & \text{otherwise} \\ [a_0, \min(3, a_1 + 1)], & T(B_{\text{cur}}) > \max(T(B_{\text{up}}), T(B_{\text{left}})). \end{cases} \quad (5)$$

Note that for the cases that one of the above and left neighboring CTBs is unavailable, i.e., the CU locates at the image boundary, full range [0, 3] is used in order to maintain the RD performance.

To evaluate the overall HR of the proposed spatial correlation-based CU depth prediction, we implemented the proposed algorithm on HM8.0, which encoded 10 test sequences with various resolutions and characteristics. Also, two sets of QPs, {22,27,32,37} and {24,28,32,36}, are tested in the coding. Table V shows the HR of four different kinds of regions, which are called boundary CTBs, smooth, normal, and complex CTBs. P_i, $i \in$[1,4], is the probability of the four kinds of regions in each sequence. HR_i, $i \in$[1,4], is the HR for the four kinds of regions. The overall HR of a sequence (HR_{ALL}) is calculated as

$$\text{HR}_{\text{ALL}} = \sum_{i=1}^{4} \text{HR}_i \cdot P_i \quad (6)$$

The HR of boundary CTBs is 100% accuracy since the full range [0,3] is adopted. The boundary CTB averagely occupies 17.38% of a frame. For the CTBs in other three kinds of regions, the average percentages P_i are 23.42%, 34.19%, and 25.37%, respectively. Correspondingly, the average HR of each region HR_i is 100%, 98.71%, 93.45%, and 95.05%, respectively. According to (6), we can get the overall HR is from 90.71% to 100%, and 96.00% on average, which is accurate enough for the depth range prediction.

The above analyses are mainly for the depth range prediction for a CTB. Actually, the depth prediction process is recursively applied at other CU levels other than largest CU (LCU). Take the CTB partition in Fig. 1 as an example, the up CU (red) and left CU (blue and green) are with 32×32, and their $D(B_{\text{up}})$ and $D(B_{\text{left}})$ are {1} and {2,3}, it can be used to predict the depth of bottom-right 32×32 CU. Furthermore, in the bottom left 32×32 CU and the up blue CU is with 16×16, and the left green CU is with 8×8, and their $D(B_{\text{up}})$ and $D(B_{\text{left}})$ are {2} and {3}, which can be used to predict the depth of bottom-right 16×16. This process can be recursively applied until the CU size is 8×8.

B. *Statistical Model-Based CU Early Termination*

The SC-CUDD approach is efficient for most regions. However, the image boundary and transitional regions between texture and smooth areas may make that approach fail due to lack of reference information and low correlation. To tackle this problem and further reduce the coding complexity, we propose the SM-CUET scheme.

In HEVC, the CU size varies from the LCU with 64×64 to the smallest CU (SCU) with 8×8. For the SCU INTRA prediction, it will further choose $2N \times 2N$ or $N \times N$ as its best PU size. In other words, SCU can be further split into four sub 4×4 blocks. Thus, we model the CU depth decision process with four level of decisions, as shown in Fig. 5. Each level of decision needs to determine whether to split or non-split. For example, it first checks 64×64 and gets its RD cost. Then, it needs to determine whether shall further split the CU

TABLE V
PROPORTION AND HR OF THE PROPOSED SC-CUDD ALGORITHM

Sequence		Resolution	Boundary CUs		Smooth		Normal		Complex		HR_{ALL}(%)
			P_1(%)	HR_1(%)	P_2(%)	HR_2(%)	P_3(%)	HR_3(%)	P_4(%)	HR_4(%)	
Set 1	Traffic	2560×1600	6.40	100.00	33.00	99.40	33.60	94.85	27.10	97.33	97.35
	Kimono	1920×1080	9.10	100.00	27.60	94.78	37.80	84.98	25.50	91.48	90.71
	Johnny	1280×720	13.40	100.00	23.10	98.80	33.30	93.65	30.20	97.89	96.97
	PartyScene	832×480	20.00	100.00	25.30	99.93	31.30	97.82	23.40	99.49	99.18
	BasketballPass	416×240	38.00	100.00	19.30	100.00	38.60	99.69	4.10	100.00	99.88
Set 2	PeopleOnStreet	2560×1600	6.40	100.00	27.60	99.35	35.20	90.30	30.90	91.30	93.82
	ParkScene	1920×1080	9.10	100.00	23.30	98.33	39.90	90.50	27.70	89.35	92.77
	FourPeople	1280×720	13.40	100.00	23.80	98.10	39.20	92.53	24.10	89.95	94.23
	BQMall	832×480	20.00	100.00	20.30	99.43	30.40	94.78	30.00	94.75	96.67
	BlowingBubbles	416×240	38.00	100.00	10.90	99.00	22.60	95.45	30.70	98.98	98.49
Average			**17.38**	**100.00**	**23.42**	**98.71**	**34.19**	**93.45**	**25.37**	**95.05**	**96.00**

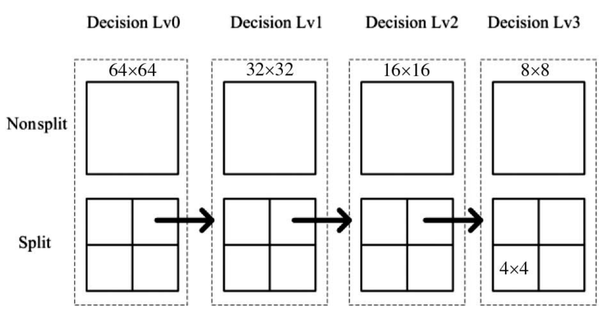


Fig. 5. Structure of four level CU decisions.

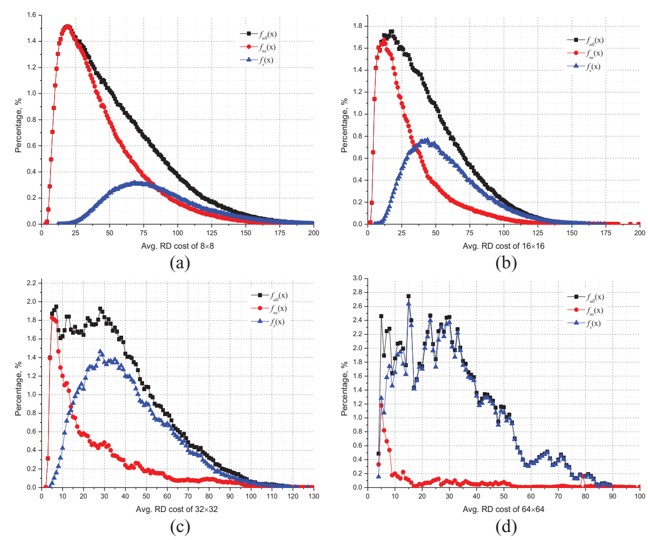


Fig. 6. PDF for four CU decision levels (*Traffic* sequence). (a) PDF of depth 3 (decision level 3). (b) PDF of depth 2 (decision level 2). (c) PDF of depth 1 (decision level 1). (d) PDF of depth 0 (decision level 0).

or not. If it is predicted as the best CU depth, the CU is not necessary to be split further and this CTB decision process is terminated. Otherwise, the CU shall be further split into four 32×32 subCUs and checked. For each of the four subCUs with 32×32, it goes to decision level 1. We will get the RD cost for each of them, and then determine whether they shall be split or not. Recursively, it ends when goes to block size with 4×4. Therefore, we have four level of decisions and we need to decide whether to spit a CU into four subCUs or not in each level of CU decision.

In the HEVC INTRA coding, the large size CUs are usually used in the smooth area while the small size CUs are used in the texture area. Thus, when we use large size CU to encode the smooth area, its RD cost usually will be smaller than that of using small size CU. In other words, if the RD cost of using large size CU is small, it is of high probability to be the best CU size. To use this property and verify our assumption, we statistically analyze the average RD cost of each depth (depth 0 to depth 3) and their probability density function (PDF). Fig. 6 shows the PDF of the four decision levels for the sequence *Traffic*. As shown in Fig. 6(a), the *x*-axis is the average RD cost at pixel level for depth 3, i.e., RD cost divided by CU size of depth 3 (8×8), and *y*-axis is the percentage of CUs. The black curve with rectangle $f_{\text{all}}(x)$ is the PDF for all 8×8 blocks, the red curve $f_{\text{ns}}(x)$ is the PDF for nonsplit 8×8 blocks, and the blue curve $f_s(x)$ is the 8×8 block that shall be further split into 4×4. We can find that CUs in depth 3 have higher probability to be nonsplit mode when it has low RD cost, where $f_{\text{ns}}(x)$ is much larger than $f_{\text{all}}(x)$ when *x* is small. Second, $f_{\text{all}}(x)$ is the sum of $f_s(x)$ and $f_{\text{ns}}(x)$. Similar observations can be found for Fig. 6(b)–(d). For smaller CU depth, e.g., 0, the $f_s(x)$ is usually larger than $f_{\text{ns}}(x)$ because most CUs shall be split and select smaller size as the optimal CU size. These PDFs satisfy

$$\begin{cases} f_s\left(x|d_i\right) &= f_{\text{all}}\left(x|d_i\right) - f_{\text{ns}}\left(x|d_i\right) \\ \int_0^{+\infty} f_{\text{all}}\left(x|d_i\right)dx &= 100\% \end{cases} \tag{7}$$

where d_i is the depth level i, $i \in 0, 1, 2, 3$. On the other hand, the percentages of nonsplit or split CUs are the cumulative density function (CDF) of $f_{\text{ns}}(x)$ and $f_s(x)$, which are

$$\begin{cases} P_{\text{ns}}\left(d_i\right) &= \int_0^{+\infty} f_{\text{ns}}\left(x|d_i\right)dx \\ P_s\left(d_i\right) = \int_0^{+\infty} f_s\left(x|d_i\right)dx. \end{cases} \tag{8}$$

Therefore, we define an RD cost threshold $T(d_i)$ for $f_s(x)$ and $f_{\text{ns}}(x)$. If the average RD cost of depth d_i is smaller than a preset threshold $T(d_i)$, the current CU can be determined as nonsplit and it is not necessary to be further split. Certainly, misclassification of spilt to nonsplit mode will be caused by using this threshold-based approach. The percentage of misclassified CUs $P_{\text{mis}}(T(d_i)|d_i)$ and additional RD cost $\Delta J_{\text{mis}}\left(T\left(d_i\right)|d_i\right)$ can be calculated as

$$\begin{aligned} P_{\text{mis}}\left(T\left(d_i\right)|d_i\right) &= \int_0^{T(d_i)} f_s\left(x|d_i\right)dx \\ \Delta J_{\text{mis}}\left(T\left(d_i\right)|d_i\right) &= \int_0^{T(d_i)} f_s\left(x|d_i\right)\left(x-x_0\right)dx \\ &\leq \int_0^{T(d_i)} f_s\left(x|d_i\right)x dx \end{aligned} \tag{9}$$

where x_0 is a positive RD cost of the CUs encoded with split mode. Meanwhile, the complexity reduction can be achieved as a number of CUs do not need to be further split. The percentage of the early terminated CUs can be calculated as

$$\begin{aligned} P_{\text{ET}}\left(T\left(d_i\right)|d_i\right) &= \int_0^{T(d_i)} f_{\text{ns}}\left(x|d_i\right)+f_s\left(x|d_i\right)dx \\ &= \int_0^{T(d_i)} f_{\text{all}}\left(x|d_i\right)dx. \end{aligned} \tag{10}$$

The CCR is direct proportional to the value of $P_{\text{ET}}(T(d_i)|d_i)$, and this $P_{\text{ET}}(T(d_i)|d_i)$ increases as $T(d_i)$ increases according to (10). In HEVC low complexity optimization, our target is to maximize the CCR subject to acceptable RD cost increase, e.g., ΔJ_T. Therefore, the optimization target is

$$\begin{aligned} &T^*\left(d_i\right) = \max P_{\text{ET}}\left(T\left(d_i\right)|d_i\right) \\ &\text{s.t. } \Delta J_{\text{mis}}\left(T\left(d_i\right)|d_i\right) \leq \Delta J_T. \end{aligned} \tag{11}$$

Since the RD cost increase $\Delta J_{\text{mis}}\left(T\left(d_i\right)|d_i\right)$ is a monotonically increasing function of $T(d_i)$, the optimal $T^*(d_i)$ can be achieved when $\Delta J_{\text{mis}}\left(T^*\left(d_i\right)|d_i\right)$ is ΔJ_T, which can be presented as

$$T^*\left(d_i\right) = {\Delta J_{\text{mis}}}^{-1}\left(\Delta J_T\right)|_{d_i}. \tag{12}$$

Now, the optimization problem of getting the optimal $T^*(d_i)$ is changed to be a prediction of $\Delta J_{\text{mis}}\left(T^*\left(d_i\right)|d_i\right)$, which has been presented in (9).

According to Fig. 6(a)–(d), we can observe that $f_{\text{all}}(x)$ and $f_{\text{ns}}(x)$ for each depth generally obey the log-normal distribution. Therefore, $f_{\text{all}}(x)$ and $f_{\text{ns}}(x)$ can be presented as

$$f_\phi\left(x|d_i\right) = \frac{1}{x\sigma_\phi\left(d_i\right)\sqrt{2\pi}} e^{\left(-\frac{\left(\ln x-\mu_\phi\left(d_i\right)\right)^2}{2\sigma_\phi^2\left(d_i\right)}\right)}, \quad \phi \in \{\text{all,ns}\} \tag{13}$$

where d_i is the depth level, $i \in 0,1,2,3$, $\mu_\phi(d_i)$ and $\sigma_\phi(d_i)$ are the mean and standard derivation of the log-normal model of CU RD cost at each d_i. In this paper, the $\mu_\phi(d_i)$ and $\sigma_\phi(d_i)$ are predicted from previous encoded frames with $\sigma_\phi(d_i) = (\ln E(X^2) - 2\ln E(X))^{1/2}$, $\mu_\phi\left(d_i\right) = 2\ln E\left(X\right) - 1/2\ln E\left(X^2\right)$, $E()$ is the mathematical expectation. Therefore, if given an acceptable RD cost increase ΔJ_T, the optimal $T^*(d_i)$ can be determined by applying (13), (9) to (12).

To verify the effectiveness and accuracy of the proposed model, we have tested and encoded various sequences with four different QPs. Fig. 7 shows the predicted and real RD cost increase, where the *x*-axis is the threshold $T(d_i)$ and *y*-axis is the normalized RD cost increase divided by CU size, i.e., ΔJ_{mis}. The legend with "Real_" prefix is the real data collected from the encoding process, the curves labeled with "Predicted_" are the predictive data from our model. We find that the real and predicted RD cost monotonically increases as the threshold $T(d_i)$ increases. Another observation is that the predicted data curves are almost overlapped with the real data curves for the four decision levels (from 64×64 to 8×8). Fig. 8 shows the RD cost for different QPs at decision level 0, i.e., determining whether split or nonsplit for CU 64×64. 1) The maximum RD cost increases as QP increases. 2) The predicted data are also very close to the real data for different QPs. To verify the prediction accuracy more precisely, the squared correlation coefficient (R^2) between the predicted and the real data for different QPs and decision levels is calculated and presented in Table VI. R^2 ranges from 0 to 1. The two data are identically the same if the R^2 equals to 1 and they are not correlated if R^2 value is 0. We observe that the R^2 value is range from 0.9580 to 0.9981, which are 0.9851 and 0.9855 on average for *Traffic* and *Kimono* sequences, respectively. It means that the real data and predicted data are highly

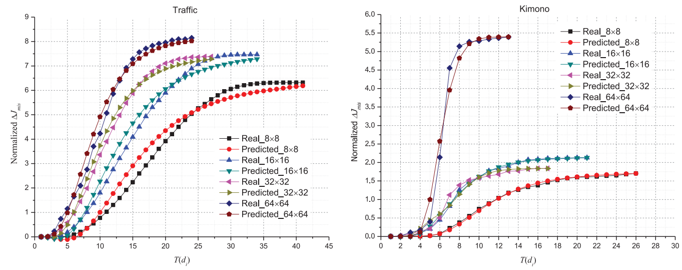


Fig. 7. Real and predictive RD cost increases for different decision levels (QP is 22).

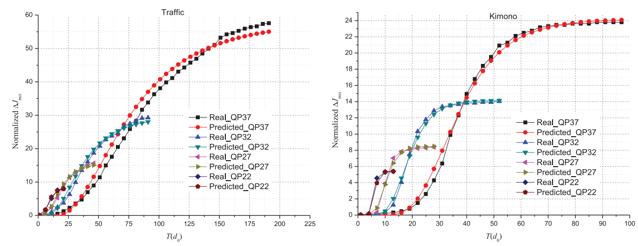


Fig. 8. Real and predictive RD cost increases for different QPs.

TABLE VI
R^2 BETWEEN THE PREDICTED AND REAL DATA FOR DIFFERENT QPS AND DECISION LEVELS

Sequence	QP	Lv0	Lv1	Lv2	Lv3	Average
Traffic	22	0.9906	0.9937	0.9863	0.9810	
	27	0.9906	0.9916	0.9865	0.9825	
	32	0.9885	0.9873	0.9841	0.9790	
	37	0.9869	0.9836	0.9777	0.9709	
	Avg.	**0.9892**	**0.9891**	**0.9837**	**0.9784**	**0.9851**
Kimono	22	0.9863	0.9736	0.9973	0.9981	
	27	0.9905	0.9768	0.9976	0.9905	
	32	0.9961	0.9852	0.9918	0.9617	
	37	0.9971	0.9842	0.9835	0.9580	
	Avg.	**0.9925**	**0.9800**	**0.9926**	**0.9771**	**0.9855**

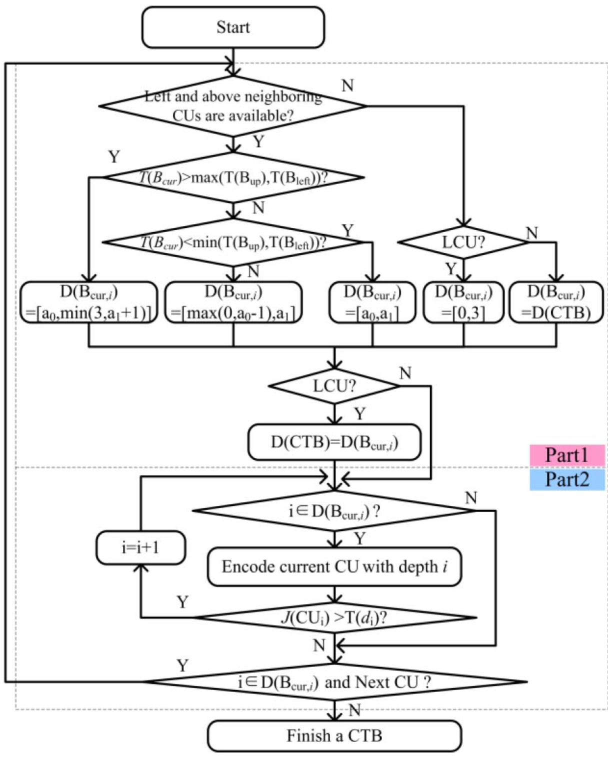


Fig. 9. Flowchart of the proposed overall CU depth decision algorithm.

correlated, and the prediction accuracy of the statistical model is sufficiently high.

To get the optimal $T(d_i)$, we shall decide the RD cost constraint ΔJ_T first. However, the RD cost ΔJ_T is not so understandable to users. Therefore, we try to map it to the allowable peak signal-to-noise ratio (PSNR) degradation, i.e., ΔPSNR, for better understandability and convenience. Basically, the RD cost ΔJ_{T} consists of distortion ΔD_T and bit rate, thus its value is larger than the distortion ΔD_T, which is

$$\Delta J_T \geq \Delta D_T = 255^2 \cdot 10^{\frac{-\mathrm{PSNR}_r}{10}} \cdot \left(1 - 10^{\frac{\Delta \mathrm{PSNR}}{10}}\right) \tag{14}$$

where PSNR_r is the PSNR of the current frame, and ΔPSNR is the target allowable PSNR degradation set by user. Since the PSNR_r is not available before coding, we predict it from the average value of previous coded frames in this work. Since the RD cost is larger than the distortion, i.e., $\Delta J_T \geq \Delta D_T$, if we use the ΔD_T to predict the optimal $T(d_i)$ (the predictive $T(d_i)$ is denoted as $T_N^*(d_i)$), this $T_N^*(d_i)$ is a little bit smaller than the real optimal $T(d_i)$ obtained from ΔJ_T, i.e., $T_N^*(d_i) = \Delta J_{\mathrm{mis}}^{-1}(\Delta D_T)|_{d_i} \leq \Delta J_{\mathrm{mis}}^{-1}(\Delta J_T)|_{d_i} = T^*(d_i)$. It means the given ΔPSNR is the upper-bound of PSNR degradation for the fast algorithm. When given a ΔPSNR, the real PSNR degradation of the proposed algorithm will be no larger than the preset ΔPSNR.

C. Proposed Overall CU Depth Decision Algorithm

Fig. 9 shows the flowchart of the proposed CU depth decision algorithm. It consists of two major parts, where part 1 is the SC-CUDD and part 2 is the SM-CUET. These two subalgorithms can be applied jointly. In the part 1 of the flowchart, D(CTB) is the depth range of the current CTB, which is predicted from the spatial neighboring CTBs and stored when checking LCU. The rest of the part 1 is the implementation of (5). Note that the $D(B_{\mathrm{cur}})$ is updated for every CU. If CU depth is within the $D(B_{\mathrm{cur}})$, the part 2 is activated for further reduce the CU depth searching. For each CU, the current CU size will be checked first. Then, based on its RD cost and the adaptive threshold $T(d_i)$, the CU is further split into four subCUs if necessary and CU depth plus 1 for further checking. Otherwise, the current CU is pruned and goes to encode next CU. Note that for the first frame, threshold $T(d_i)$ is initially set as 0 and then it will be adaptively determined based on the statistical model and different decision levels.

V. Experimental Results and Analyses

To evaluate the performance of the proposed algorithms, we implemented them, i.e., SC-CUDD, SM-CUET, and the overall algorithm, on the HEVC reference software HM8.0 [31]. Two kinds of standard test configurations, which are AIM and all INTRA main 10 (AIM10) profiles, were used in the coding experiments. The sizes of LCU and SCU are 64×64 and 8×8, respectively, which means the maximum CU depth is 4. The CTCs and settings in [32] were used. Twenty different test sequences, which are from Class A to Class E [32], and 100 frames each were encoded with four QPs, which are 22, 27, 32, and 37. For fair comparison, two benchmark schemes, Li's scheme [19] and Cho's scheme [23] (denoted by ChoCSVT), were implemented on HM8.0 and tested following the CTCs. Note that parameter α selects 0.1 for Cho's scheme and the CU depth range decision algorithm in Li's scheme was implemented for comparison. All the video coding experiments were performed on computer with CPU AMD Athlon IIX2 B24, 2.99 GHz, 2 GB memory, Windows XP operating system. Bjonteggard delta peak-signal-to-noise ratio (BDPSNR) and Bjonteggard delta bit rate (BDBR) [33] were used to evaluate the RD performance of the test schemes while compared with the original HM. Additionally, time saving (ΔT) was used to measure the CCR of the tested schemes, which is defined as

$$\Delta T = \frac{1}{4}\sum\nolimits_{i=1}^{4} \frac{T_{\mathrm{HM}}(\mathrm{QP}_i) - T_{\psi}(\mathrm{QP}_i)}{T_{\mathrm{HM}}(\mathrm{QP}_i)} \cdot 100\% \tag{15}$$

where $T_{\mathrm{HM}}(\mathrm{QP}_i)$ and $T_{\Psi}(\mathrm{QP}_i)$ are the encoding time of using the original HM and scheme Ψ with QP_i, $\Psi \in${Li, ChoCSVT, SC-CUDD, SM-CUET, overall}.

Table VII shows BDBR, BDPSNR, and time saving under different allowable RD degradation when compared with the original HM. The preset ΔPSNR varies from -0.02 to -0.20 dB with -0.02 dB step each. Five different sequences from Class A to Class E (*Traffic*, *Kimono*, *BQMall*, *BasketballPass*, and *Johnny*) were encoded with AIM encoding configuration. We observe that as the allowable RD degradation increases, the BDBR increases and the BDPSNR decreases for the test sequences. In terms of the average value over different sequences, the BDBR changes from 0.07% to 2.70% and the BDPSNR changes from 0.00 to -0.13 dB as the allowable

TABLE VII
BDBR, BDPSNR, AND TIME SAVING UNDER DIFFERENT ALLOWABLE RD DEGRADATION VALUES

ΔPSNR	Traffic (2560×1600)			Kimono (1920×1080)			BQMall (832×480)		
	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT
–0.02	0.05	0.00	16.16	0.07	0.00	21.46	0.02	0.00	12.89
–0.04	0.11	–0.01	22.14	0.14	–0.00	28.82	0.08	–0.00	18.01
–0.06	0.21	–0.01	25.45	0.21	–0.01	33.91	0.19	–0.01	21.04
–0.08	0.34	–0.02	28.63	0.73	–0.02	45.33	0.30	–0.02	23.73
–0.10	0.48	–0.02	31.04	1.43	–0.05	57.88	0.49	–0.03	25.88
–0.12	0.67	–0.03	34.10	1.38	–0.05	60.96	0.75	–0.04	27.88
–0.14	0.87	–0.04	36.28	1.68	–0.05	67.54	0.99	–0.06	29.80
–0.16	1.06	–0.05	38.30	1.53	–0.05	67.90	1.33	–0.08	31.83
–0.18	1.30	–0.06	40.43	1.64	–0.05	69.52	1.72	–0.10	34.03
–0.20	1.57	–0.08	42.32	1.51	–0.05	69.52	3.26	–0.19	41.66
	BasketballPass (416×240)			Johnny (1280×720)			**Average**		
ΔPSNR	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT
–0.02	0.01	0.00	13.09	0.20	–0.01	36.62	0.07	0.00	20.04
–0.04	0.03	–0.00	19.65	0.40	–0.02	42.60	0.15	–0.01	26.24
–0.06	0.12	–0.01	24.32	0.66	–0.03	45.40	0.28	–0.01	30.02
–0.08	0.28	–0.01	28.01	0.96	–0.04	47.70	0.52	–0.02	34.68
–0.10	0.42	–0.02	31.27	1.28	–0.05	50.28	0.82	–0.03	39.27
–0.12	0.60	–0.03	34.23	1.64	–0.06	52.12	1.01	–0.04	41.86
–0.14	0.81	–0.04	37.09	1.92	–0.07	53.87	1.25	–0.05	44.92
–0.16	1.08	–0.06	39.50	2.25	–0.08	55.75	1.45	–0.06	46.66
–0.18	1.40	–0.07	41.90	2.60	–0.10	57.38	1.73	–0.08	48.65
–0.20	4.2	–0.23	48.90	2.94	–0.11	58.48	2.70	–0.13	52.18

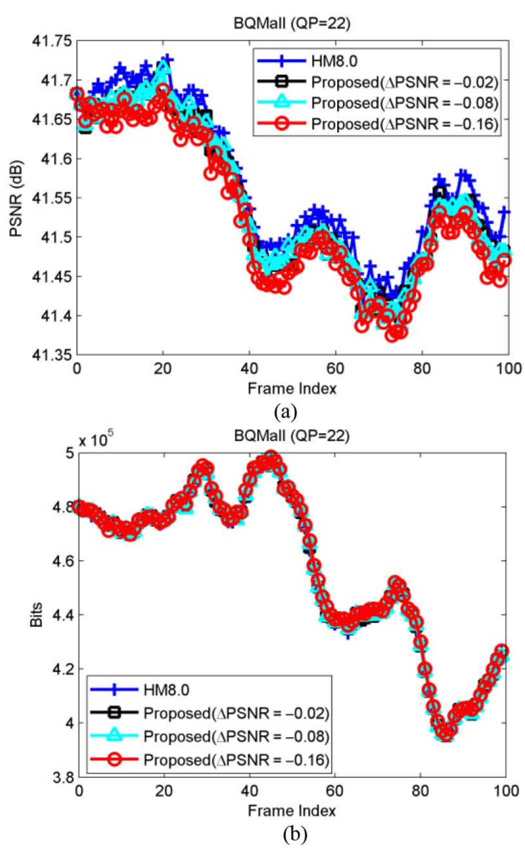


Fig. 10. Frame-by-frame PSNR and bits comparisons (BQMall). (a) PSNR. (b) Bits.

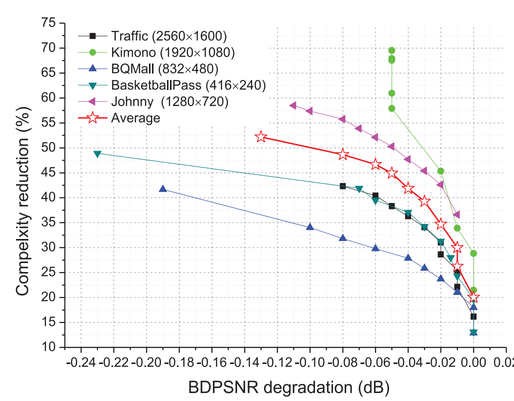


Fig. 11. Relation between BDPSNR and CCR for SM-CUET.

ΔPSNR is set from -0.02 to -0.20 dB, i.e., the RD degradation slightly increases as the target ΔPSNR increases. Second, the CCR (i.e., ΔT) of the encoder increases significantly as the ΔPSNR increases for all the test sequences. The average ΔT varies from 20.04% to 52.18%. Third, the RD performance of all the test sequence is well controlled and not larger than the preset ΔPSNR.

In addition, Fig. 10 shows frame-by-frame PSNR and bit comparisons among the original HM and SM-CUET with three different ΔPSNRs, which are -0.02, -0.08, and -0.16 dB. We can observe that the bits of the four compared schemes are very close to each other along the time. Besides, the PSNR generally decreases more as the ΔPSNRs become larger when compared with HM8.0, which conforms to the proposed statistical model. Meanwhile, the frame-by-frame PSNR difference gap is basically consistent along the time, which indicates that the proposed SM-CUET is stable along the time. Similar results can be found for other test sequences. Fig. 11 shows the relation between BDPSNR and CCR for SM-CUET for different sequences. The red curve with star symbol is the average value over the five videos. Basically, ΔT exponentially decreases as BDPSNR approaches to 0 and ΔT varies with video content properties. Usually, ΔT is relatively small for texture video and large for smooth video at the same BDPSNR cost. On the other

TABLE VIII
CODING PERFORMANCE COMPARISONS WITH AIM10 CONFIGURATION

Sets	Class	Sequences	Li's scheme [19]			Cho's scheme [23]			SC-CUDD			SM-CUET			Overall		
			BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT
Set1	Class A	Traffic	0.39	–0.02	16.81	2.22	–0.11	42.42	0.68	–0.03	36.67	0.38	–0.02	29.96	1.70	–0.08	54.83
	Class B	Kimono	0.39	–0.01	43.26	2.11	–0.07	33.51	1.27	–0.04	67.08	0.68	–0.02	46.92	2.49	–0.08	78.18
	Class C	BQMall	0.13	–0.01	13.31	1.51	–0.09	44.82	0.43	–0.02	36.59	0.22	–0.01	24.05	1.44	–0.06	51.30
		PartyScene	0.03	0.00	16.38	0.80	–0.06	35.13	0.10	–0.01	39.24	0.42	–0.02	24.51	1.26	–0.07	53.39
	Class D	BasketballPass	0.83	–0.04	7.36	1.51	–0.08	51.85	0.13	–0.01	27.99	0.42	–0.03	16.62	0.43	–0.03	43.93
	Class E	Johnny	0.84	–0.03	28.41	2.84	–0.11	59.70	0.85	–0.03	46.68	0.99	–0.04	49.62	2.15	–0.10	67.55
		Average	**0.44**	**–0.02**	**20.92**	**1.83**	**–0.09**	**44.57**	**0.58**	**–0.02**	**42.38**	**0.52**	**–0.02**	**31.95**	**1.58**	**–0.07**	**58.20**
Set2	Class A	PeopleOnStreet	0.39	–0.02	15.44	2.16	–0.11	43.37	0.60	–0.03	36.66	0.20	–0.01	25.32	1.26	–0.07	52.63
		Nebuta	0.31	–0.02	32.14	1.09	–0.07	52.85	0.65	–0.04	51.12	0.05	0.00	26.23	0.85	–0.06	62.03
		SteamLocomotive	0.21	–0.01	37.20	0.99	–0.05	27.20	0.49	–0.03	54.54	0.29	–0.02	28.85	1.23	–0.07	73.30
	Class B	ParkScene	0.35	–0.01	17.24	1.61	–0.06	40.13	0.42	–0.02	36.63	0.40	–0.02	31.06	1.21	–0.05	54.34
		Cactus	0.34	–0.01	16.42	1.95	–0.07	42.04	0.43	–0.02	37.10	0.33	–0.01	49.21	2.64	–0.08	68.93
		BQTerrace	0.40	–0.02	22.00	1.25	–0.07	45.44	0.47	–0.03	41.52	0.40	–0.01	29.10	1.44	–0.05	54.65
		BasketballDrive	1.16	–0.03	26.84	2.63	–0.06	56.56	1.27	–0.03	45.79	0.53	–0.03	32.59	1.48	–0.08	58.28
	Class C	RaceHorses	0.16	–0.01	10.83	1.74	–0.10	38.79	0.26	–0.02	33.40	0.61	–0.04	26.82	1.19	–0.07	51.79
		BasketballDrill	0.38	–0.02	11.90	1.87	–0.08	42.47	0.78	–0.03	34.48	0.15	–0.01	11.41	0.36	–0.03	46.62
	Class D	RaceHorses	0.02	0.00	8.83	1.54	–0.10	35.28	0.13	–0.01	31.34	0.67	–0.04	18.62	0.90	–0.06	43.91
		BQSquare	0.05	0.00	9.80	0.71	–0.06	43.60	0.00	0.00	32.56	0.31	–0.02	29.19	0.98	–0.05	48.53
		BlowingBubbles	0.04	0.00	8.39	1.27	–0.07	37.54	0.09	–0.01	32.30	0.16	–0.01	15.58	0.28	–0.02	43.16
	Class E	FourPeople	0.51	–0.03	15.11	2.28	–0.12	50.60	0.91	–0.05	37.53	0.94	–0.05	37.87	2.10	–0.11	60.08
		KristenAndSara	1.01	–0.05	22.90	2.40	–0.11	60.87	1.21	–0.06	43.90	0.52	–0.03	48.39	2.23	–0.11	67.82
		Average	**0.38**	**–0.02**	**18.22**	**1.68**	**–0.08**	**44.05**	**0.55**	**–0.03**	**39.21**	**0.40**	**–0.02**	**29.30**	**1.30**	**–0.07**	**56.15**
Average of all sequences			**0.40**	**–0.02**	**19.03**	**1.72**	**–0.08**	**44.21**	**0.56**	**–0.03**	**40.16**	**0.43**	**–0.02**	**30.10**	**1.38**	**–0.07**	**56.76**

TABLE IX
CODING PERFORMANCE COMPARISONS WITH AIM CONFIGURATION

Sets	Class	Sequences	Li's scheme [19]			Cho's scheme [23]			SC-CUDD			SM-CUET			Overall		
			BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT	BDBR	BDPSNR	ΔT
Set1	Class A	Traffic	0.01	0.00	16.36	0.78	–0.05	34.43	0.07	–0.01	39.2	0.7	–0.04	23.74	1.51	–0.09	52.63
	Class B	Kimono	0.4	–0.02	16.31	2.25	–0.11	41.06	0.52	–0.03	35.92	0.34	–0.02	29.18	1.46	–0.07	53.57
	Class C	BQMall	0.39	–0.01	42.26	2.04	–0.07	34.19	0.65	–0.02	63.89	0.73	–0.02	45.33	2.15	–0.07	76.05
		PartyScene	0.11	–0.01	13.42	1.54	–0.09	45.2	0.29	–0.02	36.67	0.2	–0.01	23.33	1.28	–0.06	50.2
	Class D	BasketballPass	0.8	–0.04	7.51	1.75	–0.09	51.22	0.23	–0.01	28.03	0.44	–0.03	16.16	0.48	–0.04	44.38
	Class E	Johnny	0.91	–0.03	26.76	3.16	–0.12	71.88	0.79	–0.03	45.13	0.96	–0.04	47.74	2.10	–0.09	66.23
		Average	**0.44**	**–0.02**	**20.44**	**1.92**	**–0.09**	**46.33**	**0.43**	**–0.02**	**41.47**	**0.56**	**–0.03**	**30.91**	**1.50**	**–0.07**	**57.18**
Set2	Class A	PeopleOnStreet	0.37	–0.02	15.12	2.16	–0.11	42.4	0.51	–0.03	36.31	0.18	–0.01	23.94	1.14	–0.06	51.71
		Nebuta	0.3	–0.02	31.75	1.1	–0.07	52.54	0.34	–0.02	48.51	0.04	0.00	24.92	0.55	–0.04	59.1
		SteamLocomotive	0.24	–0.01	37.28	0.98	–0.05	26.73	0.39	–0.02	52.9	0.28	–0.02	25.74	1.03	–0.06	68.39
	Class B	ParkScene	0.36	–0.01	16.87	1.61	–0.06	38.09	0.36	–0.01	36.23	0.4	–0.02	29.89	1.16	–0.05	53.72
		Cactus	0.35	–0.01	16.15	1.97	–0.07	41.16	0.41	–0.01	36.72	0.32	–0.01	47.12	2.22	–0.07	67.25
		BQTerrace	0.39	–0.02	21.99	1.29	–0.07	45.34	0.43	–0.02	41.54	0.39	–0.01	28.05	1.33	–0.05	53.85
		BasketballDrive	1.11	–0.02	25.81	2.68	–0.06	54.95	0.96	–0.02	44.78	0.49	–0.03	32.09	1.46	–0.08	58.35
	Class C	RaceHorses	0.14	–0.01	10.65	1.54	–0.09	37.34	0.18	–0.01	33.24	0.5	–0.03	26.29	1.02	–0.06	51.27
		BasketballDrill	0.33	–0.01	12.22	2.01	–0.09	41.63	0.7	–0.03	34.03	0.16	–0.01	11.13	0.34	–0.02	46.57
	Class D	RaceHorses	0.03	0.00	8.8	1.46	–0.09	35.86	0.05	0.00	31.38	0.56	–0.04	18.21	0.72	–0.05	43.99
		BQSquare	0.02	0.00	9.94	0.63	–0.05	42.9	0.05	0.00	33.34	0.28	–0.01	27.87	0.98	–0.05	47.45
		BlowingBubbles	0.02	0.00	8.45	1.25	–0.07	36.64	0.1	–0.01	32.4	0.18	–0.01	15.07	0.23	–0.01	42.92
	Class E	FourPeople	0.51	–0.03	14.75	2.38	–0.12	47.18	0.54	–0.03	36.37	0.77	–0.04	36.86	1.93	–0.10	58.41
		KristenAndSara	0.94	–0.04	21.24	2.38	–0.11	59.94	0.8	–0.04	41.67	0.44	–0.02	46.79	2.05	–0.10	66.22
		Average	**0.37**	**–0.01**	**17.93**	**1.67**	**–0.08**	**43.05**	**0.42**	**–0.02**	**38.53**	**0.36**	**–0.02**	**28.14**	**1.15**	**–0.06**	**54.94**
Average of all sequences			**0.39**	**–0.02**	**18.68**	**1.75**	**–0.08**	**44.03**	**0.42**	**–0.02**	**39.41**	**0.42**	**–0.02**	**28.97**	**1.26**	**–0.06**	**55.61**

hand, the SM-CUET is a flexible CU early termination scheme, which provides different time saving ratios according to the RD performance requirement. For example, it can give higher priority to complexity reduction for mobile video communication system when there is limitation of computation power. Also, we can give higher priority to compression efficiency for Internet-based video-on-demand applications, since the video quality under constraint bits is more essential to end users. This flexible property is important for video systems with different applications and requirements.

Besides, we also evaluate the performances of SC-CUDD and the overall algorithm. Tables VIII and IX show the coding performance comparisons with AIM10 and AIM configurations, respectively, when compared with HM. Li's scheme [19] and Cho's scheme [23] were also implemented and compared. We divide the test sequences into two sets. One is consisted of six sequences, including *Traffic, Kimono, BQMall, PartyScene, BasketballPass,* and *Johnny,* which have been used in evaluating the performance of subalgorithms SC-CUDD and SM-CUET. The other is the rest standard test sequences. The coding performances of the proposed algorithms, including SC-CUDD, SM-CUET, and the overall algorithm, are similar while encoding the video Set 1 and Set 2, which indicates that the proposed algorithms are robust.

In addition, we observe that Li's scheme reduces the complexity by 19.03% and 18.68% on an average for AIM10 and AIM configurations, respectively, compared with the original HM. Meanwhile, the average BDPSNR is -0.02 dB and the average BDBR is around 0.40%. For Cho's scheme, it can reduce the complexity by 44.21% and 44.03% on an average for the two configurations, respectively. Meanwhile, the BDPSNR is -0.08 dB and the BDBR is around 1.75%. For the SC-CUDD with AIM10 configuration, it reduces complexity by 40.16% on an average. The BDBR and BDPSNR are 0.56% and -0.03 dB, respectively, which is comparable to the original HM. We compare SC-CUDD with Li's scheme since they are, both methods, for depth range determination, and we can find that it achieves more CCR, which is $40.16\% - 19.03\% = 21.13\%$ on average. For SM-CUET, the CCR and RD performance of SM-CUET algorithm can be changed according to the requirements of the applications. In this work, we give a strict condition for ΔPSNR, which is -0.08 dB, for negligible RD degradation. We observe that the CCR is 30.10% on average and the BDPSNR is -0.02 dB on average. When combined the SC-CUDD and SM-CUET together, the BDBR and BDPSNR of the proposed overall algorithm are 1.38% and -0.07 dB, respectively, on average, which is better than Cho's scheme but a little worse than Li's scheme. However, in terms of the CCR, it achieves 43.91%–78.18%, which is 56.76% on average. In terms of the RD and CCR, the overall algorithm is almost the sum of SC-CUDD and SM-CUET, which indicates that the two algorithms are mutually complementary. The CCR achieved by the overall algorithm is $56.76\% - 19.03\% = 37.73\%$ and $56.76\% - 44.21\% = 12.55\%$ more than those of Li's and Cho's schemes. Similar results can also be found for the AIM configuration.

Since the SM-CUET is a flexible fast CU decision algorithm, the overall algorithm consisting of SM-CUET and SC-CUDD has inherited this flexibility and can also have a tradeoff between complexity and RD degradation. According to the intensive experiments and comparisons, the overall algorithm is efficient and achieves significant CCR while the RD degradation is negligible.

VI. Conclusion

In this paper, we propose a novel fast CU decision algorithm composed of two subalgorithms, including SC-CUDD and SM-CUET, to reduce the complexity of INTRA coding for mobile and industrial video applications.

1) We analyze the spatial CU depth correlation in INTRA coding, and present SC-CUDD to predict the most probable depth range for CUs based on the spatial correlation.
2) We present a statistical model and the SM-CUET algorithm, in which early termination threshold will be adaptively determined and updated based on the RD cost distribution, video content, and coding parameters.

Extensive experiments demonstrate that the proposed overall algorithm can reduce the coding complexity by 56.76% and 55.61% on an average under AIM10 and AIM configurations, respectively, which is efficient and outperforms the state-of-the-art benchmark schemes. In future, we would like to investigate learning-based approach which includes more features and sophisticated learning algorithm to make decisions better.

References

[1] C. H. Yeh, M. F. Li, M. J. Chen, M. C. Chi, X. X. Huang, and H. W. Chi, "Fast mode decision algorithm through inter-view rate-distortion prediction for multiview video coding system," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 594–603, Feb. 2014.

[2] J. Lei, J. Sun, Z. Pan, S. Kwong, J. Duan, and C. Hou, "Fast mode decision using inter-view and inter-component correlations for multiview depth video coding," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 978–986, Aug. 2015.

[3] L. Xu, S. Kwong, Y. Zhang, and D. Zhao, "Low-complexity encoder framework for window-level rate control optimization," *IEEE Trans. Ind. Electron.*, vol. 60, no. 5, pp. 1850–1858, May 2013.

[4] S. Hu, H. Wang, S. Kwong, and C.-C. J. Kuo, "Novel rate-quantization model-based rate control with adaptive initialization for spatial scalable video coding," *IEEE Trans. Ind. Electron.*, vol. 59, no. 3, pp. 1673–1684, Mar. 2012.

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

[6] T. Wiegand, G. J. 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.

[7] 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.

[8] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[9] E. Peixoto, T. Shanableh, and E. Izquierdo, "H.264/AVC to HEVC video transcoder based on dynamic thresholding and content modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 99–112, Jan. 2014.

[10] X. Shen and L. Yu, "CU splitting early termination based on weighted SVM," *EURASIP J. Image Video Process.*, vol. 2013, no. 4, pp. 1–11, Apr. 2013.

[11] H. S. Kim and R. H. Park, "Fast CU partitioning algorithm for HEVC using online learning based Bayesian decision rule," *IEEE Trans. Circuits Syst. Video Technol.*, Jun. 2015, doi: 10.1109/TCSVT.2015.2444672.

[12] J. Xiong, H. Li, F. Meng, S. Zhu, Q. Wu, and B. Zeng, "MRF-based fast HEVC inter CU decision with the variance of absolute differences," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2141–2153, Dec. 2014.

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

[14] Y. Zhang, S. Kwong, X. Wang, Z. Pan, H. Yuan, 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.

[15] 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.

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

[17] J. Vanne, M. Viitanen, and T. D. Hamalainen, "Efficient mode decision schemes for HEVC inter prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1579–1593, Sep. 2014.

[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] X. Li, J. An, and X. Guo, "Adaptive CU depth range," doc. JCTVC-E090, JCTVC of ISO/IEC and ITU-T, Geneva, Switzerland, Apr. 2011.

[20] 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.

[21] Y. Wang, X. Fan, L. Zhao, S. Ma, D. Zhao, and W. Gao, "A fast intra coding algorithm for HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP'14)*, Oct. 2014, pp. 4117–4121.

[22] M. Zhang, H. Bai, C. Lin, M. Zhang, and Y. Zhao, "Texture characteristics based fast coding unit partition in HEVC intra coding," in *Proc. Data Compress. Conf. (DCC'15)*, Apr. 2015, pp. 477–477.

[23] S. Cho and M. Kim, "Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1555–1564, Sep. 2013.

[24] Y. Kim, D. Jun, S. Jung, J. S. Choi, and J. Kim, "A fast intra-prediction method in HEVC using rate-distortion estimation based on Hadamard transform," *ETRI J.*, vol. 35, no. 2, pp. 270–280, Apr. 2013.

[25] K. Lim, J. Lee, S. Kim, and S. Lee, "Fast PU skip and split termination algorithm for HEVC intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1335–1346, Aug. 2015.

[26] H. Zhang and M. Zhan, "Fast intra mode decision for high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 660–668, Apr. 2014.

[27] N. Hu and E. H. Yang, "Fast mode selection for HEVC intra frame coding with entropy coding refinement based on transparent composite model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 9, pp. 1521–1532, Sep. 2015.

[28] L. L. Wang and W. C. Siu, "Novel adaptive algorithm for intra prediction with compromised modes skipping and signaling processes in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1686–1694, Oct. 2013.

[29] M. Jamali, S. Coulombe, and F. Caron, "Fast HEVC intra mode decision based on edge detection and SATD costs classification," in *Proc. Data Compress. Conf. (DCC'15)*, Apr. 2015, pp. 43–52.

[30] C.-S. Park, "Edge-based intramode selection for depth-map coding in 3D-HEVC," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 155–162, Jan. 2015.

[31] (2012). *HEVC Reference Software HM8.0* [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-8.0/

[32] F. Bossen, "Common test conditions and software reference configurations," JCTVC of ISO/IEC and ITU-T, JCTVC-J1100, Stockholm, Sweden, Jul. 2012.

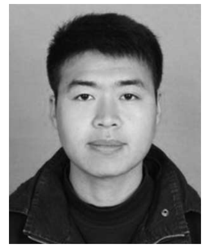
[33] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD-curves," ITU-T SG16 Q.6, VCEG-M33, Austin, TX, USA, Apr. 2001.

Yun Zhang (M'12) 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 2015, he was a Postdoctoral Researcher with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong. In 2010, he became an Assistant Professor with the Shenzhen Institutes of Advanced Technology (SIAT), CAS, where he has served as an Associate Professor since 2012. His research interests include video compression, 3-D video processing, and visual perception.

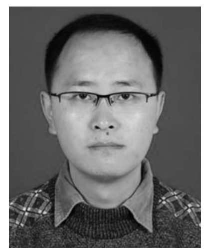
Sam Kwong (M'93–SM'04–F'13) received the B.S. degree from the State University of New York at Buffalo, Buffalo, NY, USA, in 1983; the M.S. degree from the University of Waterloo, Waterloo, ON, Canada, in 1985, both in electrical engineering; and the Ph.D. degree in electrical engineering from the University of Hagen, Hagen, Germany, in 1996.

From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada, Mississauga, Toronto, Canada. He joined Bell Northern Research Canada, Ottawa, ON, as a Member of Scientific Staff. In 1990, he became a Lecturer with the Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong, where he is currently a Professor with the Department of Computer Science. His research interests include video and image coding, and evolutionary algorithms.



Guangjun Zhang received the B.S. degree in control technology and instrument from Qufu Normal University, Rizhao, China, in 2011, and the M.S. degree in circuits and systems from Ningbo University, Ningbo, China, in 2014.

He was a Visiting Student with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China, from 2012 to 2014. Since July 2014, he has been working as a Video Codec Engineer with the Department of Cloud Platform, PPTV, Shanghai, China. His research interests include video/audio coding and optimization.



Zhaoqing Pan (S'09–M'15) received the B.S. degree in computer science and technology from Yancheng Normal University, Yancheng, China, in 2009, and the Ph.D. degree in computer science from the City University of Hong Kong, Kowloon Tong, Hong Kong, in 2014.

From June to November 2013, he was a Visiting Scholar with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA. Currently, he is a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interest includes video compression/coding.

Hui Yuan (S'08–M'12) received the B.E. and Ph.D. degrees in telecommunication engineering from Xidian University, Xi'an, China, in 2006 and 2011, respectively.

From 2011 to current, he has worked as a Lecturer, and then as an Associate Professor with the School of Information Science and Engineering, Shandong University (SDU), Jinan, China. From 2013 to 2015, he also worked as a Postdoctoral Fellow with the Department of Computer Science, City University of Hong Kong (CityU), Kowloon Tong, Hong Kong. His research interests include video coding and multimedia communication.

Gangyi Jiang (M'10) received the M.S. degree in electronics engineering from Hangzhou University, Hangzhou, China, in 1992, and the Ph.D. degree in electronics engineering from Ajou University, Suwon-si, South Korea, in 2000.

Currently, he is a Professor with the Faculty of Information Science and Engineering, Ningbo University, Ningbo, China. He has authored over 100 technical articles in refereed journals and proceedings. His research interests include video compression and multiview video coding.