Fast HEVC Inter CU Decision Based on Latent SAD Estimation

Jian Xiong, Hongliang Li, *Senior Member, IEEE*, Fanman Meng, *Member, IEEE*, Qingbo Wu, and King Ngi Ngan, *Fellow, IEEE*

***Abstract*—The emerging high efficiency video coding (HEVC) standard has improved compression performance significantly in comparison with H.264/AVC. However, more intensive computational complexity has been introduced by adopting a number of new coding tools. In this paper, a fast inter CU decision is proposed based on the latent sum of absolute differences (SAD) estimation. Firstly, a two-layer motion estimation (ME) method is designed to take advantage of the latent SAD cost. The new ME method can obtain the SAD costs for both the upper CU and its sub-CUs. Secondly, a concept of motion compensation rate-distortion (R-D) cost is defined, and an exponential model is proposed to express the relationship between the motion compensation R-D cost and the SAD cost. Then, a fast CU decision approach is designed based on the exponential model. The fast CU decision is implemented by comparing a derived threshold with the SAD cost difference between the upper and sub SAD costs. Experimental results show that the proposed algorithm achieves an average of 52% and 58.4% reductions of the coding time at the cost of 1.61% and 2% bit-rate increases under the low delay and random access conditions, respectively.**

***Index Terms*—H.264, high efficiency video coding (HEVC), inter prediction, motion estimation, video coding.**

I. Introduction

RECENTLY, with the rapid development of network communication and multimedia technologies, video contents are presented to be High Definition (HD) and Ultra HD. The resolutions of the HD videos are larger and the visual quality is better, when compared with the standard definition videos. Meanwhile, higher requirements are put forward to the video coding technologies. To meet the requirements, International Telecommunication Union-Telecom (ITU-T) and International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) jointly developed a new generation of video coding standard, namely *High Efficiency Video Coding* (HEVC).

HEVC provides significant improvement on the compression ratio in comparison with H.264/AVC [1]. Part of the improvement comes from the newly adopted quad-tree structure based *coding tree unit* (CTU) [2], which brings a flexibility of multiple sizes of coding blocks. The coding blocks are named as *coding unit* (CU). Each CU can be recursively split into four squared sub-CUs. For applications with high performance requirements, Rate-Distortion (R-D) optimization is checked in a recursive manner for all the CU sizes to obtain the optimal mode. The quad-tree coding structure can improve the coding performance significantly by the recursive RD optimization. Thus, it also brings intensive computational complexities. Reducing the coding computational complexity effectively is very important for applying HEVC on the real-time applications.

Since the process of coding a CU includes almost all the modules in the HEVC, the CU decision occupies most of coding time in HEVC. Many studies aim to design the fast CU decision algorithms for HEVC intra prediction [3]–[6] and inter prediction [7]–[9] in order to reduce the coding time. This work focuses on the fast inter CU decision. According to the employed features, the inter CU decision methods can be divided into three classes, including encoding parameters based, neighboring CU based, and R-D cost based methods.

The first class speeds up the CU decision by referring the encoding parameters, such as motion vector (MV), coded block flag (CBF), differential MV, and parameters of sample adaptive offset (SAO) [10]–[17]. For example, Ahn *et al.* proposed a fast CU decision algorithm based on the spatio-temporal encoding parameters [10], [11]. This method used the SAO parameters to measure the CU complexity, and used the MV, CBF and Transform Unit (TU) sizes to measure the motion complexity. A fast CU decision was implemented by evaluating the CU complexity and motion complexity, respectively. Correa *et al.* presented a data mining based fast CU decision method [12], [13]. The information of skip flags, merge flags, and R-D costs were extracted to generate a decision tree. Then, the CU selection was determined by the decision tree. Shen *et al.* suggested a fast Bayesian theory based CU decision algorithm [14]. They employed the sum of absolute transform differences (SATD), MV, and R-D costs as the features. The CUs were selected based on the Bayesian decision rule. In addition, Pan *et al.* developed a merge mode detection algorithm based on zero-block detection and motion estimation [15]. Kim *et al.* employed the differential MV and CBF to detect the skip mode [16]. Chiang *et al.*

Manuscript received March 30, 2015; revised August 03, 2015 and September 18, 2015; accepted October 01, 2015. Date of publication October 14, 2015; date of current version November 13, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61525102 and 61271289, and in part by the Program for Science and Technology Innovative Research Team for Young Scholars in Sichuan Province, China under Grant 2014TD0006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Adrian Munteanu.

J. Xiong, H. Li, F. Meng, and Q. Wu are with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: jxiong219@gmail.com; hlli@uestc.edu.cn; fmmeng@uestc.edu.cn; wqb.uestc@gmail.com).

K. N. Ngan is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: knngan@ee.cuhk.edu.hk).

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/TMM.2015.2491018

applied a zero-block detection using the differential MV [17]. These works efficiently reduced the encoding complexity. However, the encoding parameters are extracted in the middle of the coding process so that they are insufficient to reflect the motion compensation performance exactly.

The second class employs the depth information of the neighboring CUs to select the size of the current CU [18]–[24]. The motivation is that the depths of the neighboring CUs tend to be the same. For example, Shen *et al.* proposed a CU depth range estimation in the fast CU decision algorithm [18]. It employed the depths of the spatio-temporal neighboring CUs to estimate the depth range of the current CU. Correa *et al.* also suggested to estimate the depth of CUs using the depth of spatio-temporal neighboring CUs [19] In order to reduce the depth range, Zhang *et al.* applied the depth correlation between the spatio-temporal adjacent CTUs and the current CTUs [21]. With the purpose of skipping R-D optimization in both the frame and CU levels, Leng *et al.* employed the depth information of the neighboring CUs and the co-located CUs of the previous frame [22]. Recently, a Markov Random Field (MRF) based fast CU decision algorithm was proposed in [23], [24]. The MRF can efficiently employ the neighboring information in CU decision. In these methods, whether the optimal modes can be selected mainly depends on that the neighboring CUs are encoded with the same modes. However, around the boundary of moving objects, the neighboring CUs tend to be encoded with different modes [25].

The third class is R-D cost based fast CU decision methods [26]–[29]. Lee *et al.* proposed an R-D cost fast inter CU decision method [26]. They employed the temporary R-D cost after coding the skip mode and Inter_2N×2N PU mode. The Baysian rule on a given false ratio was proposed for the fast CU decision. Vanne *et al.* analysed the distribution of the PU modes and the relation between the modes [27]. They proposed the conditions of coding the symmetric motion partition (SMP) modes and asymmetric motion partition (AMP) modes. In order to reduce the complexity of the RD optimization, Correa *et al.* developed a fast CU decision algorithm by comparing the R-D cost with a threshold [28]. In addition, Shen *et al.* suggested a fast inter CU selection algorithm using the temporary R-D cost [29]. It is known that R-D cost is the criterion in R-D optimization of CU decision. Thus, the R-D cost based methods can exactly select the optimal modes. However, the time savings are limited since the computation of R-D costs is essentially time-consuming.

In the recent works, the common point is to propose features for early estimating the motion compensation performance. The motion compensation performance is generally measured by the sum of absolute differences (SAD) cost in motion estimation (ME). As we know, ME is an important module in the block-based hybrid video coding framework. In this module, the SAD between the original CU and the reference block is calculated to search the most matching block. The target block with the smallest SAD cost is employed as the prediction CU for coding the residuals. In conventional ME methods, the SAD is calculated only for the current CU. Actually, since the SAD calculation is a summation operation, the SAD costs of its sub-CUs are latent variables in the SAD calculation of the current CU. Here, "latent variables" are defined as variables that are inferred in the process of calculating the other variables. The latent SAD costs can efficiently reflect the motion compensation performance of the sub-CUs. Thus, this study tries to employ the latent SAD estimation for the fast CU decision. The main contributions are listed as follows.

1) We design a two-layer based ME method. The new ME method can obtain both the SAD cost of the current CU and the latent SAD costs of its sub-CUs.
2) A concept of motion compensation R-D cost is defined to describe the R-D cost related with the motion compensation. Particularly, we propose an exponential model to express the relationship between the motion compensation R-D cost and the SAD cost.
3) We propose a fast CU decision based on the latent SAD estimation. A threshold is derived based on the proposed exponential model. The fast CU decision is implemented by comparing the derived threshold with the SAD cost difference.

The rest of the paper is organized as follows. An overview of HEVC mode decision is presented in Section II. The two-layer based ME method is given in Section III. The analysis of motion compensation cost model is provided in Section IV. The proposed two-layer based fast CU decision method is proposed in Section V. Experiments are provided in Section V-C to validate the efficiency of the proposed method. Finally, we draw some concluding remarks in Section VII.

II. Overview of HEVC CU Prediction

Like H.264/AVC, HEVC [30]–[32] is still a block-based hybrid video coding standard. First, the video frames are divided into sequences of Coding Tree Units (CTUs). CTU is in a quadtree coding structure, and the nodes of the quad-tree are called as CUs. CU is the concept of the basic coding region, including a luma coding block and two chroma coding blocks. Each CU can be recursively divided into four squared sub-CUs. In HEVC reference software, the maximum and the minimum of the CU sizes are 64 × 64 and 8 × 8, respectively. Fig. 1(a) shows an example of a CTU which is recursively divided into sub-CUs, and forms a quad-tree coding structure. CUs with the sizes $2N \times 2N$, $N = 8, 16, 32$ can be recursively divided into four $N \times N$ sub-CUs. In addition, as the smallest CU, 8×8 CUs cannot be split any more.

Accordingly, prediction unit (PU) is introduced in HEVC. PU is the basic unit in the prediction. As show in Fig. 1(b), there are 8 PU modes in total. In inter prediction, the optimum mode is chosen as the one which encoded with the minimal Rate-Distortion (R-D) cost. The R-D cost is given by

$$J = D + \lambda \cdot R \tag{1}$$

where J denotes the R-D cost. The term D denotes the reconstruction distortion which is calculated as the sum of squared differences (SSD) between the original CU and the reconstructed CU. The term R denotes as the number of the coding bits. The parameter λ is the Lagrangian multiplier. This is the well-known Rate-Distortion optimization (RDO) technique.

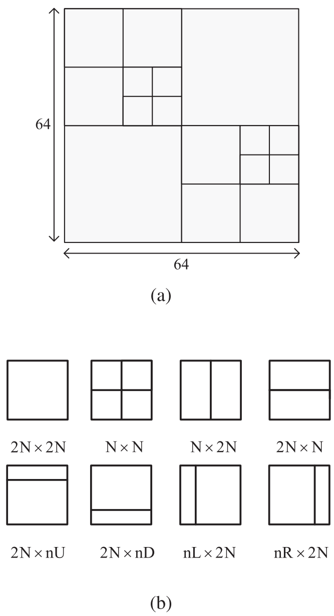


Fig. 1. HEVC coding structures. (a) The example of a CTU. (b) The PU partition modes.

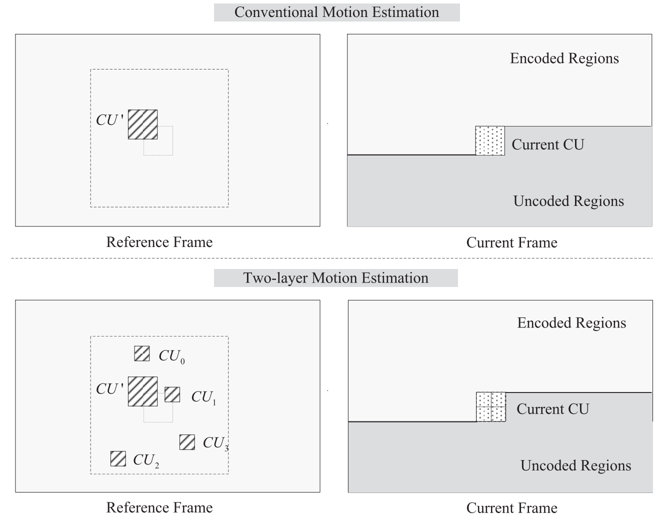


Fig. 2. Conventional motion estimation and two-layer motion estimation methods.

RDO is a time-consuming operation since it performs transform, quantization, entropy coding, inverse quantization and inverse transform.

We called a CU encoded as unsplit manner if it is encoded without dividing into sub-CUs. Otherwise, if a CU is encoded as dividing into sub-CUs, we called it encoded as split manner. For each CU (not including 8×8 CUs), R-D costs are calculated as the unsplit and split manners, respectively. The optimum split flag is determined as the one encoded with the minimal R-D cost. It is expressed by

$$s = \begin{cases} \text{unsplit}, & J_{\text{unsplit}} \leq J_{\text{split}} \\ \text{split}, & J_{\text{unsplit}} > J_{\text{split}} \end{cases} \tag{2}$$

where $s \in \{\text{unsplit}, \text{split}\}$ denotes the optimal splitting. The symbols J_{unsplit} and J_{split} denote the R-D costs of CU encoded as the unsplit and split manner, respectively.

III. Two-layer Motion Estimation

A. Conventional Motion Estimation

ME is an important module in the block-based video coding framework. Two ME algorithms are provided in the HEVC reference software, such as the full search method and the TZSearch method. The goal of both of them is to search the most matching block in the search regions of the reference frames. The target block is employed as the prediction CU for coding the residuals. The top part of the Fig. 2 shows the conventional ME algorithm. The current coded frame and its reference frame are shown on the right and left, respectively. The dashed box in the reference frame denotes the search region of the current CU. In this region, it can search a reference CU, denoted as CU'. The matching criterion is SAD cost, which is calculated as

$$J_{SAD} = SAD + \lambda_0 R_{mv} \tag{3}$$

where J_{SAD} denotes the SAD cost. The term R_{mv} denotes the estimated bits of the ME. and SAD denotes the SAD between the original CU and the reference block.

B. Two-Layer Based Motion Estimation With Latent SAD

In the conventional ME method, SAD cost is calculated only for the current CU. We called it as *Upper SAD cost*. On the contrary, we called the SAD costs of its sub-CUs as *Sub SAD costs*. However, the SAD calculation is a summation operation, i.e., the upper SAD cost is the sum of the sub SAD costs for each search point. Thus, the sub SAD costs are latent variables in the calculation of the upper SAD cost. Furthermore, as the matching criterion in the ME methods, SAD cost can reflect the motion compensation performance efficiently, i.e., the upper and sub SAD costs can reflect the motion compensation of the CU encoded as the unsplit and split manners, respectively. Therefore, we can employ both of the upper SAD cost and the latent sub SAD costs in the fast CU decision.

In order to obtain both the upper SAD cost and the latent sub SAD costs, we design a two-layer based ME methods. As shown in the bottom part of Fig. 2, the two-layer ME method also needs to search a reference block CU' for the current CU. However, the difference is that the new ME method needs to search the reference blocks of the sub-CUs. The references of the sub-CUs are denoted as CU_0, CU_1, CU_2 and CU_3, respectively.

The two-layer ME method can be easily implemented. Since the sub SAD costs are latent variables in the calculation of the upper SAD cost, only a small amount of additions, shifts and logical operations are needed. More specifically, before calculating the SAD of the current CU at each search point, it first calculates the SADs of its sub-CUs one-by-one. The summation of the sub SADs is equal to the upper SAD at each search point. Then, update the minimal upper SAD cost and the minimal sub SAD costs, respectively. After performing the ME method, we can obtain the optimal upper SAD cost and the sub SAD costs. The optimal upper SAD cost is denoted as SAD_u, which can reflect the motion compensation effect of the current CU encoded as the unsplit manner. The summation of the optimal sub SAD costs is denoted as SAD_s. It can reflect the motion compensation effect of the sub-CUs encoded as the split manner. This study tries to employ the SAD costs SAD_u and SAD_s in the fast CU decision.

IV. Motion Compensation Cost Model

In this section, we proposed a motion compensation R-D cost model to express the relationship between the SAD cost and motion compensation R-D cost. The existing rate and distortion models are introduced first. Then, a concept of motion compensation cost is presented, and an exponential model is proposed to express the motion compensation R-D cost.

A. Rate-Distortion Models

In [33], the authors have studied the R-D cost models, which include the rate model and the distortion model. The rate has been divided into two parts, the header bits and the coefficient bits, which are expressed by

$$R = R_c + R_h \tag{4}$$

where the term R_h denotes the header bits, and the term R_c denotes the coefficient bits. Furthermore, the header bits are modeled to have a good linear relationship with the numbers of the CU and PU partitions, which is given by

$$R_h = l \cdot (N_{CU} + N_{PU}) \tag{5}$$

where N_{CU} and N_{PU} denote the number of the CU and PU, respectively. However, the coefficient bits are modeled as a function of the SATD and Quantization Parameter (QP) [33]

$$R_c = \left(\frac{\sum_{i=0}^{n} SATD_i}{\sum_{i=0}^{n-1} SATD_i}\right)^{1-qcom} \cdot \frac{QP_{n-1}}{QP_n} \cdot R_{n-1} \tag{6}$$

where i is the frame number and $SATD_i$ denotes the SATD of the ith frame. The parameter n denotes the current frame number, and R_{n-1} denotes the generated source bits of the latest previous frame. The symbols QP_n and QP_{n-1} denote the QPs of the current frame and the latest previous frame, respectively. The symbol $qcom$ is a constant. Thus, the coefficient bit number of the current frame is estimated by the SATD of the previous frames, the bits number of the latest previous frame, as well as the QPs of the current frame and the latest previous frame.

The distortion model is also proposed as a function of the SATD and QP, which is an exponential formula, shown as

$$D = \alpha \cdot (SATD \cdot QP^{\gamma})^{\beta} \tag{7}$$

where α, β and γ are the model parameters. Therefore, with the trained parameters, the distortion can be estimated by the SATD and QP.

B. The Model of the Motion Compensation Cost

In order to employ SAD in the fast CU decision, we try to establish the relationship between SAD and the rate-distortion model. Combining (1) and (4), the R-D cost can be rewritten as

$$J = D + \lambda \cdot R_c + \lambda \cdot R_h. \tag{8}$$

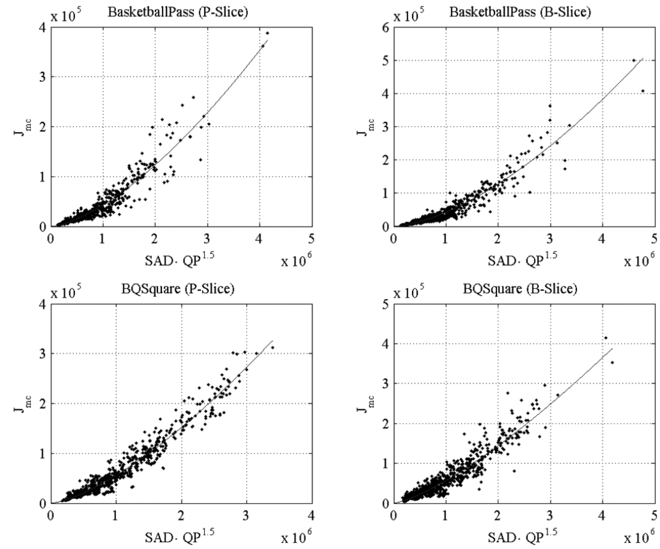


Fig. 3. Relationship between the motion compensation cost and $SAD \cdot QP^c$.

From (6) and (7), it can be observed that the distortion D and the coefficient bits R_c are both modeled as the functions of the SATD and QP. Thus, it is reasonable to combine the distortion and the coefficient bits cost into a new cost, which is defined as motion compensation cost. The symbol J_{mc} is used to denote the motion compensation cost, i.e., $J_{mc} = D + \lambda \cdot R_c$. The motion compensation cost means the cost related with the motion compensation. It has a strong relationship with the motion cost (such as SATD and SAD) and QP. Then, (8) can be rewritten as

$$J = J_{mc} + \lambda \cdot R_h. \tag{9}$$

On one hand, the motion compensation cost has a strong relationship with the motion cost and QP. On the other hand, the SAD costs of both the upper CU and its sub-CUs can be obtained in the two-layer based ME method. In order to employ the SAD costs in the fast CU decision, we analyse the relationship between the motion compensation cost, the SAD cost and QP. In [33], the parameter γ was set to 1.5. Thus, we study on the statistical data of J_{mc} and $SAD \cdot QP^c$, where $c = 1.5$. Fig. 3 shows the statistical data for P-slices and B-slices, respectively. It is interesting that there is a strong exponential relationship between J_{mc} and $SAD \cdot QP^c$, which can be expressed by

$$J_{mc} = a \cdot (SAD \cdot QP^c)^b. \tag{10}$$

This formula is called as the motion compensation cost model (MCCM). The symbols a, b and c are the model parameters. Although both of the distortion model in [33] and the proposed MCCM are in the exponential forms, MCCM is different from the distortion model. Since the values D and $SATD$ are different from J_{mc} and SAD respectively, the model parameters α and β are different with the parameters a and b.

In addition, we evaluate the fitting goodness of MCCM. Table I shows the R-square values of MCCM for the P-slice and B-slice, respectively. It can be observed that the average R-square values are above 0.997 for both the P-slice and B-slice. That is, MCCM has high accuracy on modeling the

TABLE I
FITTING GOODNESS (R-SQUARE) OF THE MOTION COMPENSATION COST MODEL

Sequences	P Slice	B Slice
BasketballPass	0.998	0.9995
BQSquare	0.9991	0.9991
BQMall	0.9958	0.9964
PartyScene	0.9965	0.9968
Average	0.9973	0.9979

relationship between the motion compensation cost and SAD cost.

V. FAST CU DECISION

In this section, a fast CU decision algorithm is proposed based on the motion compensation model. First, an analysis of CU decision is presented, and a threshold is derived to compare with the difference of the SAD costs in the fast CU decision. Then, the details of the proposed fast CU decision are provided.

A. Analysis of CU Decision

From (9), the differences between the R-D costs $J_{unsplit}$ and J_{split} can be calculated as

$$J_{\text{unsplit}} - J_{\text{split}} = J_{mc}^{u} + \lambda \cdot R_{h}^{u} - J_{mc}^{s} - \lambda \cdot R_{h}^{s} \tag{11}$$

where J_{mc}^{u} and J_{mc}^{s} are the motion compensation costs for the CU coded as the unsplit and split manners, respectively. The terms R_h^u and R_h^s denote the header bits for the CU coded as the unsplit and split manners, respectively.

Since QP is a fix value in the CU decision, (10) can be considered as the function of J_{mc} with the variation SAD cost. Performing the Taylor expansion at a point SAD_0, and ignoring the high order terms, we obtain

$$\begin{aligned} J_{mc} = a \cdot (QP^{c})^{b} \cdot SAD_0^{b} + a \cdot (QP^{c})^{b} \cdot b \cdot SAD_0^{b-1} \\ \times (SAD - SAD_0). \end{aligned} \tag{12}$$

This formula is a linear approximation of J_{mc} for the points around SAD_0.

After the two-layer based ME, the SAD costs of the upper CU and its sub-CUs can be obtained, which are denoted as SAD_u and SAD_s, respectively. From (10), the motion compensation costs J_{mc}^{u} can be estimated with the SAD cost SAD_u, and J_{mc}^{s} can be estimated with SAD_s. Then, (11) can be rewritten as

$$\begin{aligned} J_{\text{unsplit}} - J_{\text{split}} = a \cdot (QP^{c})^{b} \cdot b \cdot SAD_0^{b-1}(SAD_u \\ - SAD_s) + \lambda(R_h^u - R_h^s) \end{aligned} \tag{13}$$

Then, (2) can be rewritten as

$$s = \begin{cases} \text{unsplit}, & SAD_u - SAD_s \le T \\ \text{split}, & SAD_u - SAD_s > T \end{cases}$$
$$\text{where } T = \frac{\lambda(R_h^s - R_h^u)}{a \cdot (QP^{c})^{b} \cdot b \cdot SAD_0^{b-1}} \tag{14}$$

The symbol T denotes a threshold. It can be rewritten as

$$T = \frac{SAD_0 \cdot \lambda(R_h^s - R_h^u)}{b \cdot J_{mc_0}} \tag{15}$$

where J_{mc_0} is the corresponding motion compensation cost of SAD_0, i.e., $J_{mc_0} = a \cdot (QP^{c})^{b} \cdot SAD_0^{b}$.

B. The Proposed Fast CU Decision

There are three steps in the proposed fast CU decision, i.e., the fast skip mode detection, the fast split mode detection, and the fast unsplit determination.

1) Fast Skip Mode Detection: The skip mode is an important coding scheme in HEVC. Especially in the static regions, most of the CUs tend to be encoded with the skip mode. Furthermore, the skip mode is generally coded before the other modes. If the optimal mode is the skip mode, the other modes can be early terminated. Thus, it is necessary to detect the skip mode in the fast CU decision.

In this study, a fast skip mode detection method is proposed using the R-D costs. The R-D cost of a CU encoded with the skip mode is called as the *skip R-D cost*. When the temporary optimal mode is the skip mode after performing the skip and merge modes, the skip R-D cost is used to determine whether the skip mode is the final optimal mode. Specifically, the skip R-D costs are recorded for the last 10 CUs which are encoded with the skip mode as the optimal mode. A threshold can be estimated by these 10 R-D costs. A trimmed mean estimator [34]–[36] is used to calculate the threshold. Specifically, after discarding the maximum and minimum R-D costs in the samples, it compute the mean of the remaining R-D costs. The trimmed mean is less sensitive to outliers than the mean, such that it is a reasonable estimate of central tendency or mean for many statistical models.[1] It is expressed by

$$\widehat{J_{\text{skip}}} = \frac{1}{n-2}\left(\sum_{i=0}^{n} J_{\text{skip}_i} - \min_{i=0}^{n} J_{\text{skip}_i} - \max_{i=0}^{n} J_{\text{skip}_i}\right) \tag{16}$$

where n is the number of the sample number, i.e., $n = 10$. $J_{\text{skip}_i}, i = 0, 1, \ldots n.$ denote the R-D costs of the last 10 skip CUs. If the skip R-D cost is smaller than $\widehat{J_{\text{skip}}}$, the current CU is determined to encode with the skip mode, i.e., the other modes are early terminated. Otherwise, the other modes should be coded continuously.

2) Fast Split Mode Detection: In the fast skip mode detection, if the skip mode is not determined as the final optimal mode, the other modes should be coded continuously. The INTER_2N $\times$ 2N mode will be coded first. There will be a ME in coding this mode. As described in the above section, the two-layer motion estimation will replace the conventional ME. The new ME method can obtain both the SAD cost of the upper CU and the latent SAD costs of its sub-CUs, which are denoted as SAD_u and SAD_s, respectively. Given SAD_u and SAD_s, by estimating the value of the parameters SAD_0, J_{mc_0}, R_h^u, and R_h^s, the split flag can be determined as (14) and (15).

Since (12) is the linear approximation of J_{mc} for the points near SAD_0, it is reasonable to set SAD_0 as the average of SAD_u and SAD_s. It is expressed by

$$SAD_0 = (SAD_u + SAD_s)/2. \tag{17}$$

[1][Online]. Available: https://en.wikipedia.org/wiki/Truncated_mean

Furthermore, J_{mc_0} is estimated using the previous encoded CUs. The R-D costs of the last 10 previous CUs are recorded. For the previous CU whose SAD cost is the closest to SAD_0, the corresponding R-D cost is set as the prediction R-D cost, $\hat{J}_{pre}$. Then, J_{mc_0} is estimated as $\hat{J}_{pre}$ minus the cost of header bits. It is given by

$$J_{mc_0} = \hat{J}_{pre} - \lambda \cdot R_h^u. \tag{18}$$

Then, (15) can be rewritten as

$$T = \frac{\lambda(SAD_u + SAD_s)(R_h^s - R_h^u)}{2b(\hat{J}_{pre} - \lambda \cdot R_h^u)}. \tag{19}$$

The remain estimated parameters are the header bits R_h^u and R_h^s. The previous encoded CU is also used in the estimation. It records header bit numbers of the last 10 encoded CUs. Assume $\hat{J}_{pre}$ is larger than λR_h^s, it is easy to prove that, if R_h^s gets the maximum and R_h^u gets the minimum, T will get the maximum. It is expressed by

$$R_h^s = 4 \times \max_{i=0}^{n}\{R_{h_i}^{d+1}\}, \qquad R_h^u = \min_{i=0}^{n}\{R_{h_i}^{d}\} \tag{20}$$

where n is the sample number, i.e., $n = 10$. The terms $R_{h_i}^d$ and $R_{h_i}^{d+1}$ denote the ith header bits for the depth d and $d+1$, respectively. From (5), header bits have the linear relationship with the number of CU and PU. The number of CU and PU will be increased if it is encoded as the split manner. Then, the estimation of R_h^s is as 4 times of $\max_{i=0}^{n}\{R_{h_i}^{d+1}\}$.

Combining (19) and (20), we can obtain the maximum estimation of T. Finally, with the difference between SAD_u and SAD_s, the split flag can be early detected by the estimated T. If $SAD_u - SAD_s > T$, the coding of current INTER_2N $\times$ 2N mode will be terminated, and the current CU should be encoded as split manner. Otherwise, continue coding the other modes normally.

3) Fast Unsplit Mode Determination: In the fast split mode detection, if $SAD_u - SAD_s \leq T$, continue coding the other modes on the current depth. After that, it may get a new R-D cost, denoted as J_m, and a new header bits, denoted as k, i.e., $\hat{R}_h^u = k$. For simplicity, the new estimation of R_h^s is $4k$. The details of the fast unsplit mode determination are as follows.

1) After coding all the modes on the current depth, if the temporary optimal mode is the skip mode, it will be determined to be encoded as unsplit manner.
2) Otherwise, update J_{mc_0} with the new R-D cost J_m, i.e., $J_{mc_0} = J_m - \lambda \cdot k$, and calculate the new threshold T using the new parameters J_{mc_0}, $\hat{R}_h^u$, $\hat{R}_h^s$ as in (15). Meantime, calculate the average of the R-D costs of the previous unsplit CUs. It is expressed by

$$\widehat{J_{\text{unsplit}}} = \frac{1}{n-2}\left(\sum_{i=0}^{n} J_i - \min_{i=0}^{n} J_i - \max_{i=0}^{n} J_i\right) \tag{21}$$

where J_i is the ith R-D cost of the previous CUs which is encoded as the unsplit manner. If $SAD_u - SAD_s \leq T$ and $J_m \leq \widehat{J_{unsplit}}$, the current CU is determined to be encoded as unsplit manner.

3) Otherwise, continue coding the current CU as the split manner.

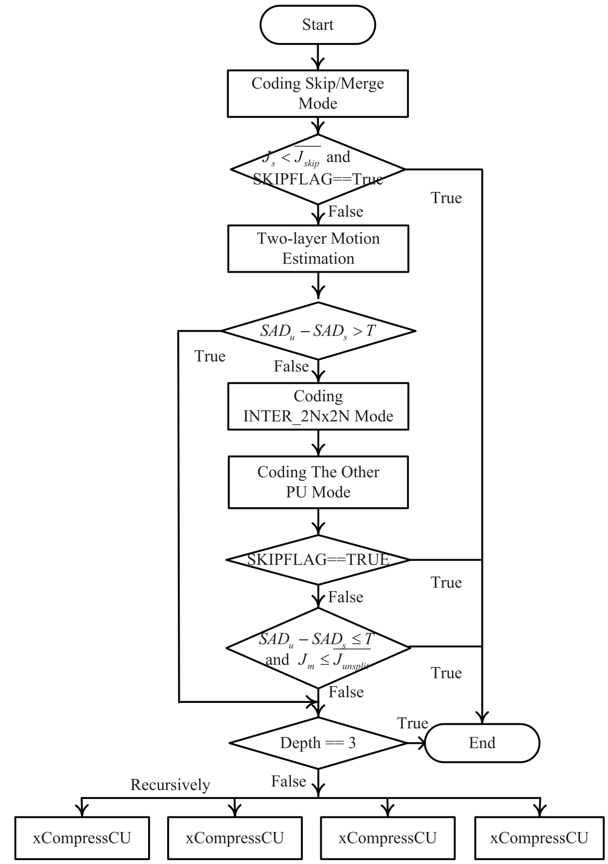


Fig. 4. Flowchart of the proposed fast CU Decision.

C. Flowchart

The flowchart of the proposed fast CU decision is shown in Fig. 4. The main steps are as follows.

1) First, code the skip and merge mode and obtain the R-D cost J_{skip}. Calculate $\widehat{J_{\text{skip}}}$ as in (16). If the temporary optimal mode is the skip mode, and $J_{skip} < \widehat{J_{\text{skip}}}$, the current CU is determined to be encoded as the skip mode, and go to step 5. Otherwise, go to step 2.
2) Perform the two-layer ME of the INTER_2N $\times$ 2N mode, and obtain the SAD costs SAD_u and SAD_s. Estimate the header bits R_h^u and R_h^s, and obtain the threshold T. If $SAD_u - SAD_s > T$, the current CU is determined to be encoded as the split manner. Otherwise, continue coding the other modes on the current depth and go to step 3.
3) If the temporary optimal mode is the skip mode, the current CU is determined to be encoded as the unsplit manner, and go to step 5. Otherwise, go to step 4.
4) Update T using the new R-D cost J_m and the new header bits. Calculate $\widehat{J_{\text{unsplit}}}$ as in (21), if $SAD_u - SAD_s \leq T$ and $J_m \leq \widehat{J_{\text{unsplit}}}$, the current CU is determined to be encoded as the unsplit manner, and go to step 5. Otherwise, code it with the split manner.
5) Finish coding the current CU on the current depth.

TABLE II
FITTED VALUES OF THE PARAMETER b IN THE MOTION COMPENSATION COST MODEL

Sequences	P Slice	B Slice
BasketballPass	1.524	1.585
BQSquare	1.447	1.412
BQMall	1.480	1.493
PartyScene	1.441	1.433
Average	1.473	1.481

VI. EXPERIMENTAL RESULTS

A. Test Conditions

In this section, the performance of the proposed fast CU decision is evaluated in terms of the difference of the computation time, the change of the Bjontegaard Delta (BD) bit-rate (BDBR) and Peak Signal to Noise Ratio (PSNR) [37]. The proposed method is integrated on the HEVC reference software, HM9.0.[2] The simulation was performed on a PC with an Intel (R) 2.30 GHz processor, 32 Gb RAM.

In the experiment, 17 sequences were tested, and Quantization Parameters (QP) values were set to 22, 27, 32, and 37 in order to cover a broad range of qualities and bit-rate. The experiment settings include the low delay ("encoder_lowdelay_P_main") and random access (Random_Access_Main). The time savings were calculated as

$$\Delta T = \frac{T_{\text{propose}} - T_{\text{reference}}}{T_{\text{reference}}} \tag{22}$$

where T_{propose} denotes the overall encoding time of the proposed fast algorithm and, $T_{\text{reference}}$ denotes the total encoding time of reference method. Since most of the CUs tend to be encoded with the large sizes, such as 64×64 and 32×32, the proposed method is implemented only on depth 0 and 1.

The model parameter b in MCCM is employed to estimate the threshold T. Table II shows the fitted values of the parameter b. It can be observed that the average fitted values are 1.473 and 1.481 for the P and B slices, respectively. Furthermore, it shows that most of the fitted values are close to 1.5. Thus, b is set to 1.5 in the experiments.

In order to validate the effectiveness of the proposed method, we compared it with the state-of-the-art fast algorithms, including the Shen's algorithms [18], [29], the Vanne's algorithm [27], the Ahn's algorithm [11] and our previous works [23], [25]. The Shen's algorithms are labeled as CUD [18] and STMD [29], respectively. Our previous works are labeled as PMD [25] and MRF [23], respectively. Since the methods [11], [18], [27], [29] were developed in different versions (HM12.0, HM2.0, HM8.0 and HM10.0, respectively) of HM reference software, we have carefully implemented them into HM9.0 reference software for fair comparison.

B. Results of the Low Delay Setting

First, we evaluated the performance of the proposed method on the low delay setting. The 8th column of Table III shows the performance of the proposed algorithm compared with the reference software. The test sequences are with different resolutions and motion intensity. From the results, it can be observed that the proposed fast CU decision algorithm can save 52% coding time of the reference software on an average. In other words, the proposed algorithm takes less than half of the coding time consumed by the reference software. Furthermore, the average BD bit-rate only increases 1.61%. In particular, for the sequence *Vidyo1*, the time saving reaches up to 74.6% with only 0.49% bit-rate increment. Fig. 5 shows the R-D curves of 6 sequences. It shows that the proposed fast CU decision algorithm can achieve a similar R-D performance as that of the reference software no matter for the low bit-rates and the high bit-rates. To sum up, the results show that the proposed algorithm can significantly reduce the coding time with only 1.61% bit-rate increase.

The 2nd column of Table III shows the results of the PMD method [25]. It can be observed that the average BD bit-rate increment of the PMD method is larger than that of the proposed method (the former is 1.89% and the latter is 1.61%). However, the proposed method can further reduce 10% coding time with respect to the PMD method. Therefore, the proposed method significantly reduce more coding time with less performance loss than the PMD method, i.e., the proposed method outperforms the PMD method.

In Table III, the 3rd column shows the results of the CUD method [18]. From the comparison results, it can be observed that the average BD bit-rate increment of the proposed algorithm is almost equal to that of the CUD method (The former is 1.61% and the latter is 1.60%). However, the proposed method can further reduce about 13% of the coding time with respect to the CUD method. In other words, the proposed algorithm reduce the coding time far more than the CUD method with almost the same coding performance degradation. Some of the results (such as *Kimono*, *BasketballDrive*) show that the proposed method achieve slightly more bit-rate increments than that of the CUD method. However, the proposed algorithm can achieve significantly more time savings than that of the CUD method. For example, 23% of the coding time can be further reduced with only 1.45% bit-rate increase for sequence *Kimono*. It can be concluded that the proposed algorithm can obtain a better trade-off in terms of the time saving and the performance degradation. The proposed algorithm outperforms the CUD method.

The 4th column of Table III shows the results of the MRF method [23]. It can be observed that the average time saving of the proposed algorithm is slightly less than that of the method [23]. However, the bit-rate increment of the proposed algorithm is less than that of the method [23] (the former is 1.61%, and the latter is 2.19%). It shows that the proposed fast CU algorithm is competitive with the method [23]. However, on one hand, the proposed method is designed with only simple operations, such as few additions and shifts. On the other hand, there are only small changes of the two-layer ME with respect to the conventional ME. Thus, the design style of the proposed method is very suitable for chip development.

The 5th, 6th, 7th columns of Table III show the results of the Ahn's algorithm [11], the Vanne's algorithm [27], the STMD method [29], respectively. As shown in Table III, the average BD bit-rate increments of these methods [11], [27], [29] are

[2][Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn HEVCSoftware/tags/HM-9.0rc2/

TABLE III
PERFORMANCE COMPARISON OF PROPOSED METHOD WITH RECENT WORKS (LOW DELAY)

Sequences	PMD [25]		CUD [18]		MRF [23]		Ahn [11]		Vanne [27]		STMD [29]		**Proposed Method**	
	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
Traffic	2.04	-46.7	1.72	-48.0	2.81	-66.6	1.67	-45.3	1.03	-46.0	1.02	-52.0	1.88	-60.2
PeopleOnstreet	1.91	-42.3	1.23	-36.9	1.31	-50.9	1.44	-29.2	0.76	-34.5	0.95	-35.2	1.11	-41.2
BasketballDrive	3.69	-52.3	1.76	-39.7	3.52	-62.6	1.14	-41.4	1.27	-40.4	0.74	-43.2	2.22	-56.6
Cactus	1.90	-41.4	2.24	-45.4	3.15	-60.3	1.87	-46.0	1.20	-39.7	0.87	-45.1	1.65	-55.5
ParkScene	1.80	-43.7	0.99	-41.3	2.51	-62.6	1.38	-50.3	0.86	-44.4	0.86	-48.5	2.09	-55.2
Kimono	1.76	-48.2	0.48	-30.2	2.58	-66.1	0.97	-45.0	1.04	-36.7	0.16	-37.6	1.93	-53.2
Vidyo1	1.65	-57.0	1.78	-64.7	2.97	-75.8	1.16	-64.5	1.30	-55.5	0.69	-62.4	0.49	-74.6
Vidyo3	1.22	-51.6	0.95	-55.8	2.83	-70.5	1.01	-57.5	1.35	-49.0	1.33	-58.2	0.49	-66.5
Vidyo4	2.17	-60.7	2.78	-57.7	2.75	-73.1	1.12	-56.4	1.04	-51.0	-0.09	-60.1	0.77	-69.7
BasketballDrill	1.27	-35.3	2.84	-37.6	2.20	-57.9	1.58	-37.2	0.62	-37.4	0.50	-37.3	0.96	-48.3
BQMall	3.21	-48.1	5.72	-43.8	1.96	-59.6	1.27	-42.2	0.79	-40.2	1.24	-41.7	1.70	-52.6
RaceHorsesC	2.47	-33.0	0.93	-28.1	1.59	-48.9	1.13	-26.8	0.73	-33.1	0.80	-30.4	1.29	-40.9
PartyScene	1.48	-31.7	0.53	-30.6	1.18	-48.0	0.99	-31.2	0.55	-31.4	0.95	-30.3	2.47	-40.3
BasketballPass	1.73	-39.7	0.51	-34.9	1.31	-61.6	0.84	-40.7	0.57	-38.3	1.14	-35.4	3.30	-53.1
BlowingBubbles	1.73	-31.3	0.79	-24.1	2.11	-50.4	1.17	-32.7	1.19	-35.0	1.31	-30.3	2.82	-42.6
RaceHorses	0.92	-29.7	0.44	-21.2	1.24	-44.7	0.97	-23.0	1.06	-30.1	1.41	-25.4	1.46	-30.3
BQSquare	1.25	-23.2	1.60	-26.1	1.21	-54.7	1.30	-32.6	0.78	-43.0	1.42	-41.8	0.75	-43.6
Average	1.89	-42.1	1.60	-39.2	2.19	-59.7	1.24	-41.3	0.95	-40.3	0.90	-42.1	1.61	-52.0
SD	0.70	10.4	1.31	12.3	0.77	9.1	0.28	11.5	0.26	7.1	0.42	11.1	0.81	11.6

TABLE IV
PERFORMANCE COMPARISON OF PROPOSED METHOD WITH RECENT WORKS (RANDOM ACCESS)

Sequences	PMD [25]		CUD [18]		MRF [23]		Ahn [11]		Vanne [27]		STMD [29]		**Proposed Method**	
	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
Traffic	3.51	-52.2	2.05	-51.4	3.65	-71.3	1.62	-63.6	0.95	-51.0	0.94	-59.9	2.20	-69.5
PeopleOnstreet	1.99	-37.7	2.47	-39.0	2.06	-54.6	1.08	-28.1	1.47	-46.3	1.12	-38.0	1.47	-46.3
BasketballDrive	3.31	-45.2	6.62	-48.7	3.93	-65.7	2.34	-53.2	1.99	-43.7	0.95	-48.4	3.82	-61.3
Cactus	2.16	-43.8	5.87	-47.6	3.66	-64.9	2.67	-54.9	1.12	-44.9	0.85	-52.4	2.40	-62.4
ParkScene	3.13	-48.2	1.86	-44.2	2.78	-61.7	1.49	-51.8	0.87	-49.3	0.99	-56.5	2.05	-64.8
Kimono	2.06	-36.3	4.67	-39.1	3.98	-68.6	1.84	-59.8	1.29	-40.9	0.73	-41.5	4.93	-57.8
Vidyo1	2.03	-58.7	1.96	-61.4	2.90	-78.5	1.69	-54.5	1.32	-54.5	0.56	-69.3	1.21	-77.7
Vidyo3	1.92	-53.5	1.48	-57.5	3.17	-75.4	0.83	-55.2	0.96	-52.6	0.79	-66.0	0.80	-73.1
Vidyo4	2.61	-56.3	2.80	-59.6	2.99	-76.3	2.13	-58.6	1.28	-53.4	0.62	-66.4	1.32	-74.8
BasketballDrill	1.52	-44.1	4.76	-41.5	3.27	-60.1	1.63	-40.7	0.97	-41.6	0.74	-41.0	1.40	-52.1
BQMall	1.71	-44.6	6.11	-45.5	2.66	-63.6	1.97	-44.3	0.92	-44.9	0.83	-46.4	1.88	-58.3
RaceHorsesC	1.76	-32.4	3.20	-32.1	2.94	-51.3	2.43	-34.6	1.52	-36.4	1.34	-31.8	2.06	-44.2
PartyScene	1.50	-32.9	1.58	-36.4	1.45	-53.1	0.83	-36.4	0.57	-37.6	0.96	-37.9	2.09	-47.0
BasketballPass	1.08	-47.7	2.99	-38.9	1.85	-66.2	1.53	-32.1	0.60	-48.0	1.11	-49.2	0.60	-59.7
BlowingBubbles	1.27	-31.0	0.58	-27.6	2.12	-53.3	0.93	-36.5	0.66	-39.9	0.97	-35.5	2.67	-50.1
RaceHorses	0.87	-30.5	1.02	-23.3	1.96	-46.9	1.27	-27.9	1.00	-34.1	1.08	-26.8	1.56	-34.2
BQSquare	1.21	-39.6	0.51	-37.1	1.15	-62.3	0.84	-47.2	0.51	-46.7	0.59	-49.7	1.58	-58.8
Average	1.98	-43.2	2.97	-43.0	2.74	-63.2	1.59	-45.8	1.06	-45.0	0.89	-48.1	2.00	-58.4
SD	0.77	8.9	1.95	10.7	0.85	9.2	0.59	11.6	0.39	6.1	0.21	12.5	1.06	11.8

slightly less than that of the proposed method. However, the proposed method significantly reduce more coding time (11%, 12% and 10%, respectively) than these methods [11], [27], [29]. The above experimental results indicate that the proposed method is superior or comparable to the three state-of-the-art fast CU decision methods on the low delay setting.

C. Results of the Random Access Setting

We also evaluate the proposed algorithm on the random access setting. The 8th column of Table IV shows the results of the proposed method when compared with the reference software. It can be observed that the proposed method can reduce 58.4% of the coding time with only 2% BD bit-rate increment. The results indicate that the proposed method can speed up the CU decision effectively.

The 2nd column of Table IV shows the results of the PMD method [25]. As shown in Table IV, the PMD method reduce average 43.2% of the total coding time with 1.89% BD bit-rate increment. The proposed method reduce more (15%) coding time than PMD with only 0.11% BD bit-rate increment. It can be concluded that the proposed method significantly outperforms the PMD method on the random access setting.

The 3rd column of Table IV shows the results of the CUD method. The CUD method can reduce 43% of the coding time with 2.97% BD bit-rate increment in average. The proposed method can further reduce about 15% of the total coding time with respect to the CUD method. Furthermore, the average bit-rate increment of the proposed method is less than that of the CUD method. It can be concluded that the proposed method significantly outperforms CUD in terms of both the time saving and coding performance degradation on this setting.

The 4th column of Table IV shows the results of the MRF-based method [23]. It can be observed that, the time saving of the proposed method is slightly smaller than that of the method [23] (the former is 58.4%, the latter is 63.2%). However, the bit-rate increment of the proposed method is significantly lower than that of the method [23] (The former is 2%, the latter is 2.78%). On this setting, the proposed method can obtain a better

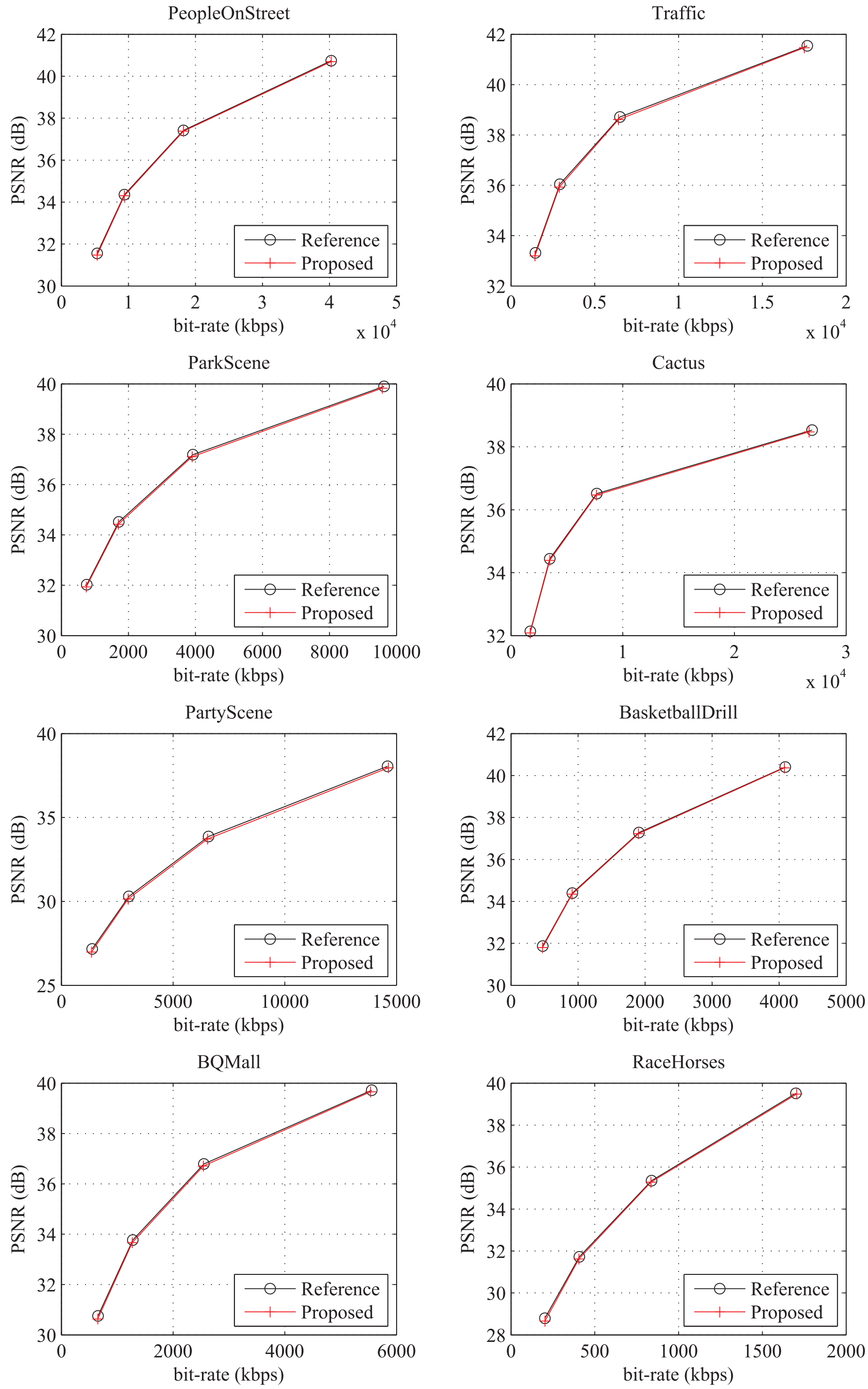


Fig. 5. R-D performance of the proposed fast CU decision.

trade-off between the time saving and coding performance loss than the method [23].

The 5th, 6th, 7th columns of Table IV show the results of the Ahn's algorithm [11], the Vanne's algorithm [27], the STMD method [29], respectively. Furthermore, the performance comparisons of the random access setting are similar to that of the low delay setting. On this setting, it can be observed that the BD bit-rate increment of the methods [11], [27], [29] are slightly less than that of the proposed method, and the SDs are smaller than that of the proposed method. However, the proposed method saves more coding time (13%, 13% and 10%, respectively) than these methods [11], [27], [29]. It can be concluded that the proposed method is superior and comparable to the state-of-the-art algorithms on this setting.

We also calculate the standard deviations (SD) of the BD bit-rate increments and time savings. As shown in Table III, and Table IV, for the proposed method, the SDs of bit-rate increment are only 0.81% and 1.06% under the low delay and random access settings, respectively. It can be observed that the proposed method is robust to different video contents. On one hand, the

TABLE V
PERFORMANCE OF THE SUB-ALGORITHMS (LOW DELAY)

Sizes	Sequences	FSkip		FSplit		FUnsplit		Overall	
		BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
Class A (2560×1600)	Traffic	0.45	-39.2	0.50	-9.8	0.86	-49.4	1.88	-60.2
	PeopleOnstreet	0.22	-15.7	0.37	-22.2	0.81	-24.0	1.11	-41.2
Class B (1920×1080)	BasketballDrive	0.53	-26.5	1.87	-11.6	0.81	-45.8	2.22	-56.6
	Cactus	0.34	-29.4	0.59	-14.4	0.90	-42.1	1.65	-55.5
	ParkScene	0.42	-33.4	0.34	-9.1	1.14	-44.4	2.09	-55.2
	Kimono	0.63	-20.6	2.34	-17.1	0.91	-40.1	1.93	-53.2
Class E (1280×720)	Vidyo1	0.24	-55.1	0.54	-6.5	0.22	-62.2	0.49	-74.6
	Vidyo3	-0.21	-46.6	0.33	-7.9	0.15	-56.1	0.49	-66.5
	Vidyo4	0.22	-49.3	0.67	-5.8	0.26	-61.1	0.77	-69.7
Class C (832×480)	BasketballDrill	0.28	-22.3	0.55	-15.9	0.58	-35.3	0.96	-48.3
	BQMall	0.43	-28.5	0.41	-15.8	0.89	-38.1	1.70	-52.6
	RaceHorsesC	0.27	-13.3	0.33	-18.8	0.76	-25.6	1.29	-40.9
	PartyScene	0.43	-14.6	0.09	-16.1	1.88	-27.1	2.47	-40.3
Class D (416×240)	BasketballPass	0.42	-23.1	0.16	-5.7	2.27	-35.0	3.30	-53.1
	BlowingBubbles	0.63	-18.1	0.52	-11.8	1.76	-31.1	2.82	-42.6
	RaceHorses	0.29	-8.3	0.24	-12.5	0.84	-18.0	1.46	-30.3
	BQSquare	0.03	-33.6	0.18	-12.9	0.36	-38.8	0.75	-43.6
Average		0.33	-28.1	0.59	-12.6	0.91	-39.7	1.61	-52.0

TABLE VI
PERFORMANCE OF THE SUB-ALGORITHMS (RANDOM ACCESS)

Sizes	Sequences	FSkip		FSplit		FUnsplit		Overall	
		BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
Class A (2560×1600)	Traffic	0.58	-50.2	0.83	-9.2	1.16	-57.4	2.20	-69.5
	PeopleOnstreet	0.20	-21.6	0.67	-25.0	1.04	-30.9	1.47	-46.3
Class B (1920×1080)	BasketballDrive	1.27	-36.4	3.27	-12.6	0.94	-49.3	3.82	-61.3
	Cactus	0.56	-40.1	1.77	-13.6	1.20	-49.2	2.40	-62.4
	ParkScene	0.54	-44.5	0.74	-9.5	1.23	-53.2	2.05	-64.8
	Kimono	1.18	-29.3	3.28	-16.9	1.15	-44.6	4.93	-57.8
Class E (1280×720)	Vidyo1	0.23	-62.7	1.44	-5.8	0.58	-67.5	1.21	-77.7
	Vidyo3	0.09	-56.0	0.95	-7.1	0.45	-63.4	0.80	-73.1
	Vidyo4	0.62	-57.2	2.14	-6.2	0.29	-65.7	1.32	-74.8
Class C (832×480)	BasketballDrill	0.38	-29.8	1.35	-15.7	0.76	-39.9	1.40	-52.1
	BQMall	0.38	-36.7	0.70	-15.2	1.20	-44.4	1.88	-58.3
	RaceHorsesC	0.84	-18.0	1.33	-20.1	1.09	-28.6	2.06	-44.2
	PartyScene	0.57	-25.0	0.40	-15.0	1.47	-34.8	2.09	-47.0
Class D (416×240)	BasketballPass	-0.01	-41.3	0.40	-11.8	0.40	-46.5	0.60	-59.7
	BlowingBubbles	0.54	-33.0	0.42	-13.1	1.02	-37.3	2.67	-50.1
	RaceHorses	0.24	-13.1	0.44	-15.6	0.95	-21.2	1.56	-34.2
	BQSquare	0.42	-40.2	0.15	-4.4	1.22	-49.6	1.58	-58.8
Average		0.51	-37.4	1.19	-12.8	0.95	-46.1	2.00	-58.4

bit-rate increment SDs of the proposed method are close to that of the PMD and MRF-based methods, and they are smaller than that of the CUD method. That is, the proposed method achieves comparable or superior robustness as the PMD, MRF, and CUD methods. On the other hand, the bit-rate increment SDs of the proposed method are larger than that of the Ahn's algorithm [11], the Vanne's algorithm [27], the STMD method [29]. However, the proposed method results in significantly more time savings. Furthermore, it can be observed that the SDs of the time savings are around 10% for all the comparative methods. It indicate that the time savings are mainly influenced by the video contents. When encoding the simple videos, more time is likely to be saved. On the contrast, less time is likely to be saved for the complex videos.

D. Contributions of the Sub-Algorithms

As introduced in the above section, the proposed method includes three mode detection algorithms, i.e., fast skip mode detection (FSkip), fast split mode detection (FSplit), and fast unsplit detection (FUnsplit). In order to investigate the performance contribution of each part, we have evaluated the BD bit-rates and time savings of these sub-algorithms. Experimental results of the low delay and random access settings are shown in Tables V and VI, respectively.

As shown in Table V, FSkip yields about average 28.1% reduction of the total coding time with only 0.33% BD-rate increment. Especially for the sequence *Vidyo1*, FSkip greatly reduces about 55.1% coding time with only 0.24% BD bit-rate increment. On the other hand, FUnsplit saves about 39.7% coding time with only 0.91% BD bit-rate increment. The sequence *Vidyo1* also achieves the best performance in all the tested sequences. That is, the sequence *Vidyo1* achieves the most time saving for both the FSkip and FUnsplit detections. The reason is that, the sequence *Vidyo1* is with static background and smooth motion. It makes that CUs in this sequence are more likely to be encoded as the skip mode and the unsplit manner. The results of the other sequences also validate that FSkip and FUnsplit reduce more coding time for the sequences with smooth motion than that of the complex sequences. Furthermore, FSplit results in about 12.6% time saving with average 0.59% BD bit-rate increment. It can be observed that, for *Vidyo1*, FSplit saves almost the least coding

TABLE VII
TIME SAVINGS OF DIFFERENT VIDEO RESOLUTIONS

Conditions	Resolutions				
	Class A 2560×1600	Class B 1920×1080	Class C 832×480	Class D 416×240	Class E 1280×720
Low Delay	-50.67 %	-55.14 %	-45.52 %	-42.39 %	-70.28
Random Access	-57.87 %	-61.57 %	-50.39 %	-50.71 %	-75.22
Average	-54.27 %	-58.35 %	-47.96 %	-46.55 %	-72.75

TABLE VIII
RESULTS OF TIME SAVINGS ON DIFFERENT QPS

Conditions	QP			
	QP=37	QP=32	QP=27	QP=22
Low Delay	-62.86%	-55.38%	-48.65%	-41.20%
Random Access	-68.42%	-61.70%	-54.99%	-48.31%
Average	-65.64%	-58.54%	-51.82%	-44.75%

TABLE IX
PERFORMANCE OF THE PROPOSED METHOD FOR THE AVC TESTED SEQUENCES

Sizes	Sequences	Low Delay		Random Access	
		BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
QCIF	Akiyo	0.16	-65.4	0.94	-70.7
	City	3.71	-44.1	1.69	-45.3
	Crew	0.67	-34.8	1.27	-39.1
	Soccer	0.70	-29.7	0.42	-31.4
CIF	Tempete	2.75	-37.5	2.46	-45.8
	Bus	2.28	-33.4	2.46	-42.0
	Forman	2.62	-49.0	2.49	-54.1
	Paris	1.54	-48.7	1.78	-55.4
Average		1.80	-42.8	1.69	-48.0

time. However, for the sequences with high intensity motion (such as, *RaceHorsesC*, *PartyScene* and *PeopleOnstreet*), FSplit saves significantly more coding time than the sequences with smooth motion (such as, *Vidyo1*, *Vidyo3*, and *BQSquare*). It can be considered that FSplit is complementary with FUnsplit. The overall scheme which combines these sub-algorithms together yields average 52% time saving with only 1.61% BD bit-rate loss, i.e., the overall scheme saves the most coding time. Results in Table VI show that the performance of random access setting is similar with that of the low delay setting. It is observed that all the sub-algorithms reduce the total coding time effectively with acceptable BD bit-rate increments.

E. Time Saving Statistics and Analyses

We also investigate the performance of the proposed method tested with different parameters, such as QP, and video resolutions. Table VII shows the time saving of the proposed method performed on different resolutions. From the results, we can see that the average time saving of class E reaches up to 72.75%, which is significantly larger than that of the other classes. That is because the videos in class E are particular. Backgrounds of these video are static, and the foreground are with small motions. For these videos, most of the CUs are encoded as the unsplit manner, or the skip mode. Furthermore, the time savings of the other classes are around 50%. It indicates that the proposed algorithm is robust to a wide range of video resolutions.

Table VIII shows the time savings of different QPs on the two settings, respectively. When QP is set to 37, the time saving is 66.64%. When QP is set to 22, the time saving is 44.75%. It can be observed that the time savings of high QPs are larger than that of the low QPs. When a video is encoded with the high QP, the CUs tend to be encoded as the unsplit manner, and most of the CUs encoded as the skip mode. It indicates that the proposed method is more efficient for the low bit-rate applications. However, even for the QP 22, almost half of the coding time is reduced. It can be concluded that the proposed method is robust to a wide range of bit-rates.

F. Results of the AVC Tested Sequences

In order to validate that the proposed algorithm is generally applicable to different contents, the AVC sequences with sizes QCIF and CIF are tested. Results are shown in Table IX. It can be observed that the proposed algorithm reduces average 42.8% and 48.0% of the total coding time, when the average BD bit-rate increments are only 1.80% and 1.69% for the low delay and random access settings, respectively. Especially for the sequence *Akiyo*, the proposed method saves up to 65.4% and 70.7% coding time with only 0.16% and 0.94% BD bit-rate increments for the low delay and random access settings, respectively. The results indicate that the proposed algorithm is also applicable to the QCIF and CIF sequences.

G. Performance of Applying the Full Search ME

The TZsearch ME method is applied in this work. Some of the search points are skipped in the TZsearch ME. In this case, SAD calculation of the sub-CU only covers the points that the CU ME reaches. However, the sub-CUs ME may search on different points. In order to investigate the differences, we tested the proposed scheme with the full search ME on the low delay setting. Experimental results are shown in Table X. It can be observed that, when applying the full search ME, the average BD bit-rate increment is slightly less than that of applying the TZsearch ME (The former is 1.64%, and the latter is 1.84%). However, both of the coding performance losses are significantly small, i.e., the proposed algorithm is suitable to both of the two ME schemes.

In addition, we investigate the encoding time of different schemes. As shown in Table X, T_{full} and T_{tz} denote the encoding time consumed by the HM reference software when applying the full search ME and TZsearch ME, respectively. The symbols $T_{\text{prop+full}}$ and $T_{\text{prop}+tz}$ denote the encoding time of the proposed fast CU decision algorithm when applying the full search ME and TZ search ME, respectively. The relative time savings of applying the two ME schemes are denoted as $\frac{T_{\text{prop+full}}-T_{\text{full}}}{T_{\text{full}}}$ and $\frac{T_{\text{prop}+tz}-T_{tz}}{T_{tz}}$, respectively. The relative time saving of applying the full search ME is larger than that of applying the TZsearch ME (The former is about 60.6%, and the latter is about 44.0%), i.e., more modes are early detected or skipped when applying the full search ME. It can be concluded that the more accurate SAD calculation of the sub-CUs will lead to more time saving. However, when comparing with

TABLE X
PERFORMANCE OF APPLYING THE FULL SEARCH ME AND THE TZSEARCH ME

Sequences	Full Search		TZsearch		
	BDBR (%)	$\frac{T_{prop+full}-T_{full}}{T_{full}}$ (%)	BDBR (%)	$\frac{T_{prop+tz}-T_{tz}}{T_{tz}}$ (%)	$\frac{T_{prop+tz}-T_{full}}{T_{full}}$ (%)
BasketballDrill	0.59	-66.9	0.96	-48.3	-97.2
BQMall	1.55	-71.9	1.70	-52.6	-97.7
RaceHorsesC	1.30	-54.0	1.29	-40.9	-95.3
PartyScene	2.39	-56.2	2.47	-40.3	-96.2
BQSquare	2.63	-60.4	3.30	-43.6	-97.2
BlowingBubbles	2.97	-58.6	2.82	-42.6	-96.6
RaceHorses	1.06	-43.0	1.46	-30.3	-94.6
BasketballPass	0.65	-73.9	0.75	-53.1	-98.0
Average	1.64	-60.6	1.84	-44.0	-96.6

T_{full}, the average time saving of applying the TZsearch ME is significantly larger than that of applying the full search ME (The former is about 96.6%, and the latter is about 60.6%). The reason is that the full search ME drastically increase the computational complexity. Therefore, it is better to implement the proposed fast CU decision algorithm on the encoder with the fast ME methods, such as the TZsearch ME. The future work is improving the accuracy of the sub-CU SAD calculation.

VII. CONCLUSION

In this paper, a two-layer ME based fast CU decision method has been proposed. This method employs the latent SAD estimation for the fast CU decision. First, we design a two-layer based ME method. The new ME method obtains the SAD costs for both the upper CU and its sub-CUs. The upper SAD cost can reflect the motion compensation effect of the upper CU, as well as the sub SAD costs can reflect the motion compensation effect of the sub-CUs. Then, we define the concept of motion compensation R-D cost and its exponential expressing model. This model can express the relationship between the motion compensation R-D cost and the SAD cost. Based on the exponential model, we can derive a threshold. The fast CU decision is converted to the problem of estimating the threshold. The split flags can be early determined by comparing the estimated threshold with the SAD cost difference. Experimental results show that the proposed algorithm significantly reduce average 52% and 58.4% encoding time with only 1.61% and 2% bit-rate increases for the low delay and random access settings, respectively.

REFERENCES

[1] T. Wiegand, G. 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.

[2] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, and W.-J. Han, "Hm7: High efficiency video coding (HEVC) test model 7 encoder description," in *Proc. 9th JCT-VC Meeting*, Apr. 2012, pp. 6–11, Art. ID JCTVC-I1002.

[3] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. Consumer Electron.*, vol. 59, no. 1, pp. 207–213, Feb. 2013.

[4] 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.

[5] J. Xiong and H. Li, "Fast and efficient prediction unit size selection for HEVC intra prediction," in *Proc. Int. Symp. Intell. Signal Process. Commun. Syst.*, Nov. 2012, pp. 366–369.

[6] J. Xiong, "Fast coding unit selection method for high efficiency video coding intra prediction," *Opt. Eng.*, vol. 52, no. 7, 2013, Art. ID 071504.

[7] H. L. Tan, F. Liu, Y. H. Tan, and C. Yeo, "On fast coding tree block and mode decision for high-efficiency video coding (HEVC)," in *Proc. IEEE Int. Confe. Acoust., Speech Signal Process.*, Mar. 2012, pp. 825–828.

[8] 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.

[9] L. Shen, P. An, Z. Zhang, Q. Hu, and Z. Chen, "A 3D-HEVC fast mode decision algorithm for real-time applications," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 11, no. 3, pp. 34–56, Jan. 2015.

[10] S. Ahn, M. Kim, and S. Park, "Fast decision of CU partitioning based on SAO parameter, motion and PU/TU split information for HEVC," in *Proc. Picture Coding Symp.*, Dec. 2013, pp. 113–116.

[11] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatio-temporal encoding parameters for HEVC inter coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 422–435, Mar. 2015.

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

[13] G. Correa, P. Assuncao, L. Agostini, and L. da Silva Cruz, "A method for early-splitting of HEVC inter blocks based on decision trees," in *Proc. 22nd Eur. Signal Process. Conf.*, Sep. 2014, pp. 276–280.

[14] X. Shen, L. Yu, and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," in *Proc. Picture Coding Symp.*, May 2012, pp. 453–456.

[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] J. Kim, J. Yang, K. Won, and B. Jeon, "Early determination of mode decision for HEVC," in *Proc. Picture Coding Symp.*, May 2012, pp. 449–452.

[17] P.-T. Chiang and T. S. Chang, "Fast zero block detection and early CU termination for HEVC video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 1640–1643.

[18] 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.

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

[20] J.-H. Lee, C.-S. Park, B.-G. Kim, D.-S. Jun, S.-H. Jung, and J. S. Choi, "Novel fast PU decision algorithm for the HEVC video standard," in *Proc. 20th IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1982–1985.

[21] Y. Zhang, H. Wang, and Z. Li, "Fast coding unit depth decision algorithm for interframe coding in HEVC," in *Proc. Data Compression Conf.*, Mar. 2013, pp. 53–62.

[22] J. Leng, L. Sun, T. Ikenaga, and S. Sakaida, "Content based hierarchical fast coding unit decision algorithm for HEVC," in *Proc. Int. Conf. Multimedia Signal Process.*, May 2011, vol. 1, pp. 56–59.

[23] 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.

[24] J. Xiong, H. Li, F. Meng, B. Zeng, S. Zhu, and Q. Wu, "Fast and efficient inter CU decision for high efficiency video coding," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 3715–3719.

[25] 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.
[26] J. Lee, S. Kim, K. Lim, and S. Lee, “A fast CU size decision algorithm for HEVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 411–421, Mar. 2015.
[27] J. Vanne, M. Viitanen, and T. Hamalainen, “Efficient mode decision schemes for HEVC inter prediction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1579–1593, Sep. 2014.
[28] M. Cassa, M. Naccari, and F. Pereira, “Fast rate distortion optimization for the emerging HEVC standard,” in *Proc. Picture Coding Symp.*, May 2012, pp. 493–496.
[29] L. Shen, Z. Zhang, and Z. Liu, “Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1709–1722, Oct. 2014.
[30] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[31] K. Ugur *et al.*, “High performance, low complexity video coding and the emerging HEVC standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1688–1697, Dec. 2010.
[32] W.-J. Han *et al.*, “Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.
[33] S. Ma, J. Si, and S. Wang, “A study on the rate distortion modeling for high efficiency video coding,” in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 181–184.
[34] T. J. Rothenberg, F. M. Fisher, and C. B. Tilanus, “A note on estimation from a Cauchy sample,” *J. Amer. Statist. Assoc.*, vol. 59, no. 306, pp. 460–463, 1964.
[35] D. Bloch, “A note on the estimation of the location parameter of the Cauchy distribution,” *J. Amer. Statist. Assoc.*, vol. 61, no. 315, pp. 852–855, 1966.
[36] P. J. Rousseeuw and S. Verboven, “Robust estimation in very small samples,” *Comput. Statist. Data Anal.*, vol. 40, no. 4, pp. 741–758, 2002.
[37] *Calculation of Average PSNR Differences Between RD Curves*, ITU-T SC16/Q6, VCEG-M33, Apr. 2001.

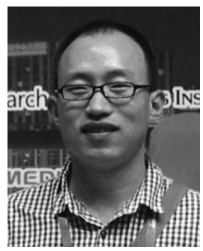
Jian Xiong received the B.Sc. degree in computer science and technology from Anhui University of Finance and Economics, Bengbu, China, in 2007, the M.Sc degree in computer application technology from Xihua University, Chengdu, China, in 2010, and the Ph.D. degree in single and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2015.

From February 2014 to May 2014, he was a Research Assistant with the Image and Video Processing Laboratory, The Chinese University of Hong Kong, Hong Kong. Since September 2015, he has been a Lecturer with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include image and video coding, quality evaluation, and perceptual video coding and processing.

Hongliang Li (SM’12) received the Ph.D. degree in electronics and information engineering from Xian Jiaotong University, Xi’an, China, in 2005.

From 2005 to 2006, he was a Research Associate with the Visual Signal Processing and Communication Laboratory (VSPC), Chinese University of Hong Kong (CUHK), Hong Kong. From 2006 to 2008, he was a Postdoctoral Fellow with the VSPC, CUHK. He is currently a Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored numerous technical articles in international journals and conferences. He is a co-editor of the book *Video Segmentation and Its Applications* (Springer, 2011). His research interests include image segmentation, object detection, image and video coding, visual attention, and multimedia communication system.

Dr. Li is a Member of the Editorial Board of the *Journal on Visual Communications and Image Representation*, and the Area Editor of *Signal Processing: Image Communication*. He served as a Technical Program Co-Chair in ISPACS 2009, General Co-Chair of the ISPACS 2010, Publicity Co-chair of the IEEE VCIP 2013, Local Chair of the IEEE ICME 2014, and TPC Member in a number of international conferences, such as ICME 2013, ICME 2012, ISCAS 2013, PCM 2007, PCM 2009, and VCIP 2010. He serves as a Technical Program Co-Chair for IEEE VCIP 2016. He was selected as the New Century Excellent Talent in University, Chinese Ministry of Education, China, in 2008.



Fanman Meng (S’11–M’13) received the Ph.D. degree in single and information processing from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2014.

From July 2013 to July 2014, he was a Research Assistant with the Division of Visual and Interactive Computing, Nanyang Technological University, Singapore. He is currently an Associate Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored numerous technical articles in international journals and conferences. His research interests include image segmentation and object detection.

Prof. Meng is a Member of the IEEE Circuits and Systems Society. He was the recipient of the Best Student Paper Honorable Mention Award at the 12th Asian Conference on Computer Vision 2014 and the Top 10% Paper Award at the IEEE International Conference on Image Processing 2014.

Qingbo Wu received the B.E. degree in education of applied electronic technology from Hebei Normal University, Shijiazhuang, China, in 2009, and is currently working toward the Ph.D. degree in electronic engineering at the University of Electronic Science and Technology of China, Chengdu, China.

From February 2014 to May 2014, he was a Research Assistant with the Image and Video Processing Laboratory, The Chinese University of Hong Kong, Hong Kong. He is currently a Visiting Student with the Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include image/video coding, quality evaluation, and perceptual modeling and processing.

King Ngi Ngan (F’00) received the Ph.D. degree in electrical engineering from Loughborough University, Loughborough, U.K., in 1982.

He is currently a Chair Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong. He was previously a Full Professor with the Nanyang Technological University, Singapore, and the University of Western Australia, Crawley, WA, Australia. He has been appointed Chair Professor with the University of Electronic Science and Technology, Chengdu, China, under the National Thousand Talents Program, since 2012. He holds honorary and visiting professorships at numerous universities in China, Australia, and Southeast Asia. He has authored or coauthored 3 books, 7 edited volumes, and over 370 refereed technical papers, and has edited 9 special issues in journals. He holds 15 patents in the areas of image/video coding and communications.

Prof. Ngan is a Fellow of the IET and IEAust, and was an IEEE Distinguished Lecturer from 2006 to 2007. He served as Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the *Journal on Visual Communications and Image Representation*, the *EURASIP Journal of Signal Processing: Image Communication*, and the *Journal of Applied Signal Processing*. He was Chair or Co-Chair for a number of prestigious international conferences on image and video processing, including the 2010 IEEE International Conference on Image Processing, and served on the Advisory and Technical Committees of numerous professional organizations.