An Efficient Fast Mode Decision Method for Inter Prediction in HEVC

Jinlei Zhang, Bin Li, *Member, IEEE*, and Houqiang Li, *Senior Member, IEEE*

***Abstract*— The emerging High Efficiency Video Coding (HEVC) standard adopts many advanced techniques with flexible combinations, which enables HEVC to achieve about 50% bit-rate reduction for similar perceptual video quality relative to the prior video coding standard H.264/Advanced Video Coding. However, the enormously increased encoding complexity of HEVC inevitably becomes one of the greatest challenges for real-time applications. Among all the factors resulting in the increase in encoding complexity of HEVC, the quad-tree structure for coding units (CUs) with different sizes and accordingly a large number of prediction modes is one critical reason. Thus, it is greatly desired to develop a fast mode decision method for HEVC to reduce the computational complexity. In this paper, considering that HEVC employs the quad-tree structure, and the distortion of each sub-CU can indicate whether the current mode is suitable for current CU, we explore the relationship between the impossible modes and the distribution of the distortions to help the encoder skip checking the unnecessary modes. Besides, since the residual values can reflect the prediction result directly, we propose a method to skip some motion estimation operations according to the distribution of the residuals. Experimental results show that the proposed method can save about 77% of encoding time with only about a 4.1% bit-rate increase compared with HM16.4 anchor, while compared with the fast mode decision method adopted in HM16.4, the proposed algorithm can save about 48% of encoding time with only about a 2.9% bit-rate increase.**

***Index Terms*— Coding unit (CU), fast mode decision, High Efficiency Video Coding (HEVC), inter prediction, motion estimation (ME).**

I. Introduction

The H.264/Advanced Video Coding (AVC) standard [1] is currently the predominant video coding standard and has been widely used by a variety of applications. When H.264/AVC was under standardization, only a few specific devices supported high-definition (HD) videos. However, with the fast development of multimedia techniques, HD videos are now supported by more and more electronic devices and higher resolutions begin to attract the market's attention. High Efficiency Video Coding (HEVC) [2] standard, which is a successor to H.264/AVC, is designed to meet the emerging demands of high-quality video services, especially for HD TV. According to [3], HEVC can provide approximately a 50% bit-rate reduction for equivalent perceptual quality relative to the performance of H.264/AVC (especially for HD videos). However, the high compression performance mainly results from the adoption of several advanced techniques, which enormously increases the complexity of the encoder. Among all the factors, the quad-tree coding structure is one of the most critical reasons.

Manuscript received February 17, 2015; revised May 22, 2015; accepted July 7, 2015. Date of publication August 3, 2015; date of current version August 5, 2016. This work was supported in part by the 973 Program under Contract 2013CB329004, in part by the Natural Science Foundation of China under Contract 61390510 and Contract 61325009, and in part by the Fundamental Research Funds for the Central Universities under Contract WK2100060011. This paper was recommended by Associate Editor M. Rabbani. *(Corresponding author: Houqiang Li.)*

J. Zhang and H. Li are with the Chinese Academy of Sciences Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China (e-mail: jlzhang@mail.ustc.edu.cn; lihq@ustc.edu.cn).

B. Li is with Microsoft Research Asia, Beijing 100080, China (e-mail: libin@microsoft.com).

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/TCSVT.2015.2461991

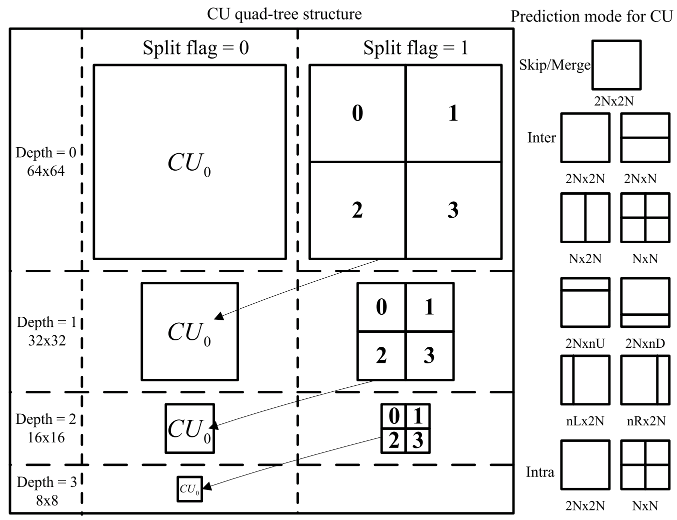


Fig. 1. CU quad-tree structure in HEVC and the prediction mode for CU in each depth.

In HEVC, as shown in Fig. 1, the basic processing unit is largest coding unit (LCU) which is nonoverlapped squared block. And each coding unit (CU) can be recursively divided into four partitions with quad-tree representation unless the CU is smallest CU (SCU). The typical size of the LCU is 64×64 with depth 0 and the size of the SCU is 8×8 with depth 3. For each CU, it can be further divided into prediction units (PUs) which is the basic unit to carry prediction information [e.g., motion vector (MV), intra-prediction direction] in HEVC. Different from CUs whose shape must be square, PUs can be rectangle. The HEVC encoder enables 10 different PU types, including Inter_2N $\times$ 2N, Inter_2N $\times$ N, Inter_N $\times$ 2N, Inter_N $\times$ N, the asymmetric modes (Inter_2N $\times$ nU, Inter_2N $\times$ nD, Inter_nL $\times$ 2N, and Inter_nR $\times$ 2N), and intra modes (Intra_2N $\times$ 2N and Intra_N $\times$ N). In the full search algorithm in inter prediction, all the prediction modes are searched one by one in each depth,

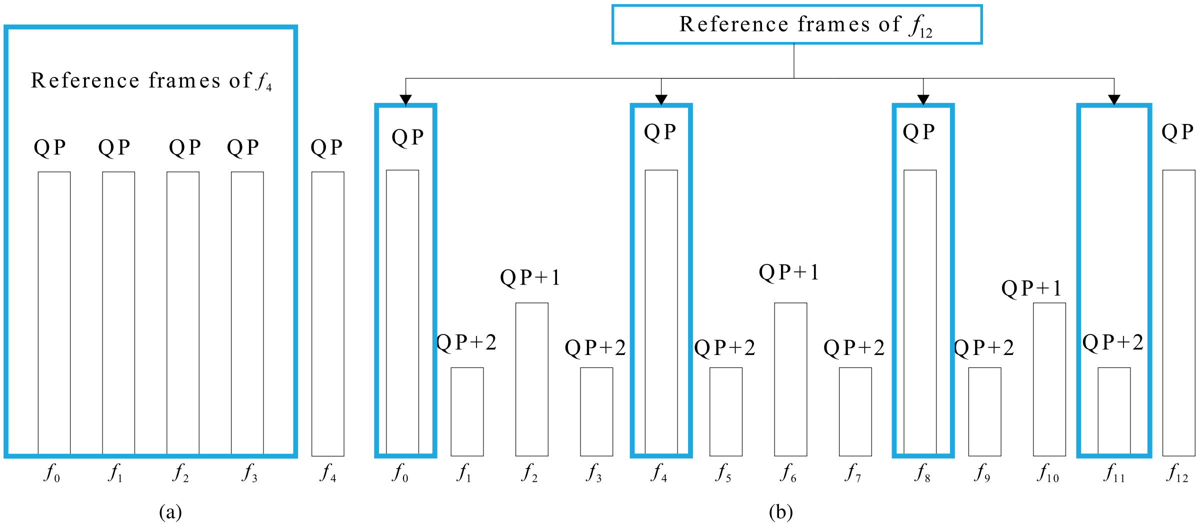


Fig. 2. Example of the typical low-delay reference structure. (a) Flat QP setting. (b) Hierarchical QP setting.

except that Inter_N × N (except Inter_4 × 4) and Intra_N × N are only for SCU and the asymmetric modes are not for 8 × 8 CU.

As shown in Fig. 1, there are inter-prediction and intra-prediction modes in HEVC. In fact, there have already been some works [4], [5] on fast intra prediction for HEVC. However, similar to H.264/AVC, the most time-consuming computation in HEVC is the inter-prediction operations with motion estimation (ME). Therefore, in this paper, we focus on fast inter prediction. For an inter frame, the full-search algorithm has to check all the possible MV positions exhaustively for all the possible modes to decide the best PU mode. As a result, the full-search algorithm achieves the best prediction performance with the highest computational complexity. Although many fast ME algorithms [6], [7] have been explored, the computation for ME is still high. Thus, the full-search algorithm is unwelcome in real-time applications. In addition, such computation-consuming operation is a critical bottleneck for mobile applications, since a great amount of energy will be required to perform video coding tasks.

Extensive investigations have been done to reduce the computational complexity for H.264/AVC, aiming to get a tradeoff between compression performance and encoding complexity. One kind of these methods only chooses some most possible modes as candidates [8]–[14] by exploring the modes' correlation and rate distortion costs' (RD-costs) correlation in temporal and spatial neighboring macroblocks (MBs), and others try to save the ME time [15] by inferring the motion information from the first reference frame to the others. However, as shown in Fig. 1, in the quad-tree structure of HEVC, each CU is recursively divided into four sub-CUs in a top-down order, which makes the number of prediction modes for each LCU be far more than the number of modes for each MB in H.264/AVC. Besides, as shown in Fig. 2, H.264/AVC usually uses flat quantization parameter (QP) while HEVC specifies the hierarchical coding structure. Thus, the fast mode decision methods designed for H.264/AVC cannot be applied to HEVC straightforwardly, which will be discussed in detail in Section II.

In fact, there have already been some works on fast inter prediction for HEVC. Most of the fast inter-prediction researches for HEVC [16]–[20] are designed for the quad-tree coding structure, which aim to skip some unnecessary mode selections on some depths using the spatial and temporal depth correlation [16]–[19] or save the ME time by merging the neighboring small PUs with the same MVs into large PUs or CUs [20]. Besides, several approaches [21], [22] are proposed to early determine SKIP or Merge mode according to the RD-cost or all-zero residual blocks. In addition, some proposals [23]–[25] have been adopted into the HEVC reference software, which propose to skip some PU processes or mode selections in some depths based on the residuals and MV difference (MVD) information. The combination of these proposals outperformed many previous fast algorithms designed for HEVC thanks to the obvious time saving with negligible performance loss and its simplicity. However, these methods have not removed the computational redundancy efficiently and the complexity of these existing methods are still high for real-time applications, since almost all of these algorithms only save 50% of the encoding time.

Although the inter-mode decision in HEVC is very complicated, there remains some redundancy in the mode selection operations. On the one hand, some information can help the encoder to eliminate some unnecessary modes. For example, the distribution of the distortions can indicate the property of the contents in the picture, which helps the encoder to decide whether the current CU size is suitable for the contents. On the other hand, in the quad-tree coding structure, the encoder repeats to perform ME for each CU under different PU partitions. Since the residual values can reflect the prediction accuracy directly, the encoder can skip some unnecessary ME operations based on the residual distribution. Thus, we try to propose some fast mode decision methods taking advantage of these information. In this paper, we have two main contributions.

1) Different from the existing methods which concentrate on achieving the most possible modes according to the RD-cost threshold or the spatial and temporal

neighboring blocks' modes, we propose to determine the impossible modes based on the distribution of the distortions so that we can speed up the encoding process with small performance loss. We obtain a correlation between the distortion and the prediction mode, which helps us define the conditions that the CUs can be partitioned into sub-CUs directly.

2) Different from the existing algorithms which focus on the idea of skipping the ME operation for CUs, we use the distribution of the residuals of each PU to skip some ME operations not only for CUs but also for PUs.

In summary, we propose a fast mode decision algorithm for inter prediction in HEVC fully utilizing the information obtained from CUs to skip some mode selections and ME operations for PUs or sub-CUs.

The rest of this paper is organized as follows. In Section II, we give a summarization of the related works in H.264/AVC and HEVC. In Section III, we first present three proposed fast mode decision algorithms in detail, and then outline the overall algorithm. The experimental results are shown in Section IV with corresponding discussions. Finally, we conclude this paper in Section V with a summary of the proposed algorithm.

II. Related Work

A. Fast Mode Decision Algorithms Proposed for H.264/AVC

There have been many investigations on fast mode decision for inter prediction in H.264/AVC, which can be classified into three categories. In this section, we review these methods one by one and analyze why they cannot be applied to HEVC straightforwardly.

The first class tries to find a way to decrease the multiple reference frame ME time. Based on the assumption that the content is in the state of uniform motion in a straight line, Su and Sun [15] present a fast multiple reference frame ME method which infers the MVs of the other reference frames from the MV of the first reference frame. In this way, no extra ME process is needed for the other reference frames, which helps to decrease the encoder's computational complexity. However, the methods of this class are unsuitable for the widely accepted coding structures in HEVC which are defined in common test conditions (CTCs) and provide good compression performances. We take the low-delay case as the example to explain the problem. The algorithm proposed in [15] is designed for the reference frame structure in H.264/AVC, as shown in Fig. 2(a), which uses the nearest reconstructed frames as the reference frames. However, as shown in Fig. 2(b), which uses one nearest reconstructed frame and three high-quality reconstructed frames as the reference frames. As reported in [26] and [27], this reference frame structure can provide a significant performance gain. Thus, it will be widely used for different applications. Therefore, we choose this reference frame structure as a typical example. From Fig. 2(a) and (b), we can see that the distance between the reference frames in hierarchical QP setting in HEVC is larger than that in flat QP setting and the assumption cannot be satisfied easily. As a result, it is inaccurate to derive the MVs of the high-quality reference frames from the first reference frame.

The second class selects some modes as candidates by making full use of the spatial and temporal correlation. Ren *et al.* [8] generate the histogram of the spatial and temporal neighboring MBs' modes, and chooses the candidates from them so that there is no need to search all the possible modes. Lee and Park [9] propose a fast mode decision method based on motion cost and intra-prediction cost. It classifies the MBs into homogeneous regions and heterogeneity regions. Furthermore, for the homogeneous regions, it selects three candidates according to the motion cost and the intra-prediction cost, which helps to save the time of RD-cost computation and motion compensation for the other modes. As we know, the possible modes of each LCU are far more than that of one MB in H.264/AVC. Thus, adoption of such scheme directly for HEVC may lead to inefficiency in exploiting the correlation of the modes between neighboring LCUs due to the larger basic CU. In HEVC, as discussed in Section I, assuming that the size of LCU is 64×64 and the size of SCU is 8×8, there are six possible modes (SKIP, Inter_2N $\times$ 2N, Inter_N $\times$ 2N, Inter_2N $\times$ N, Intra_2N $\times$ 2N, and Intra_N $\times$ N) for each SCU and nine possible modes (SKIP, Inter_2N $\times$ 2N, Inter_N $\times$ 2N, Inter_2N $\times$ N, Inter_2N $\times$ nU, Inter_2N $\times$ nD, Inter_nL $\times$ 2N, Inter_nR $\times$ 2N, and Intra_2N $\times$ 2N) for the other CUs. Therefore, for one LCU, the number of the possible modes is

$$\mathrm{num}_{\mathrm{HEVC}} = ((6^4+9)^4+9)^4+9 \approx 7.1 \times 10^{49}. \tag{1}$$

As a result, the relationship between RD-cost (or motion cost) and modes cannot be explored simply by the statistic methods. And the relationship of the neighboring blocks' modes is too complex to explore. Therefore, the methods proposed in [8] and [9] and the similar ideas [10], [11] cannot be easily applied to HEVC.

The last class [12], [13] aims to estimate a threshold of current picture's RD-cost using the correlation among the adjacent pictures' RD-costs, and early determine the best mode depending on the threshold. The RD-cost function is defined as

$$J = \mathrm{SSD} + \lambda R \tag{2}$$

where SSD is the sum of squared difference between the reconstructed pixels and the original pixels in current CU, R is the number of bits used to code these pixels' information, and λ is the Lagrangian multiplier. Most algorithms of this type have been known to be useful for H.264/AVC configurations that deal with low-delay P (LDP) coding structure with flat QP. However, HEVC usually uses the hierarchical QP setting, as shown in Fig. 2(b), which makes the quality and bit-rate of neighboring frames vary significantly. Therefore, in HEVC, the neighboring frames do not have the similar RD-cost values. Although we can use the RD-costs of the frames which belong to the same hierarchical level with the current frame to estimate the threshold, the temporal distance between these frames is relatively large such that the context in the collocated position may have changed. Therefore, the RD-cost of the temporal collocated CU is not accurate enough to estimate the RD-cost of the current CU. In conclusion, these methods based on the neighboring frames' RD-cost correlation are not appropriate for HEVC.

Besides, Rhee *et al.* [28] summarize many fast mode decision algorithms for H.264/AVC, and modifies the previous algorithms for HEVC. As shown in [28], the RD performance drop caused by these previous algorithms is quite large for HEVC. Although the algorithm proposed in [28] provides an obvious time reduction with negligible RD performance loss, it is only designed for the pipeline architecture of P picture only test configuration. In summary, the main ideas of fast mode decision used in H.264/AVC are unsuitable for HEVC due to the flexible CU structure and the reference frame structure of HEVC.

B. Fast Mode Decision Algorithms Designed for HEVC Inter Prediction

1) Fast Mode Decision Researches for HEVC: First, we overview the algorithms designed for the quad-tree coding structure of HEVC.

1) The algorithm proposed in [16] derives the depth range of each LCU according to the depth information of its spatial neighboring LCUs and temporal collocated LCU, and then only does mode selection for the depth in the range instead of for all depths. Besides, it proposes three early stop conditions. The first one decides whether current CU belongs to the homogeneous area based on the motion information. The second one gets the predicted RD-cost of current CU according to the RD-cost of the collocated CU. The last one is similar to [25] which uses early SKIP detection scheme.
2) The research in [21] also proposes three strategies: 1) early SKIP detection using the spatial–temporal and inter-level correlation; 2) early determine CU sizes according to the neighboring CUs; and 3) early determine the best mode according to the predicted RD-cost. As we discussed in Section II-A, the algorithms based on RD-cost is not efficient for hierarchical QP setting, as the results shown in [21], the third method can only save about 15% of encoding time to prevent obvious performance loss.
3) The work of [17] uses the similar idea of [16], both of them are based on the main idea of skipping some mode selections on unnecessary depth. Different from [16] and [17] is adaptive to different applications with different encoding complexity requirement. It sets a target of the encoder complexity first, and then determines the largest depth of each LCU depending on two factors, i.e., the maximum depth of the LCUs at the same position in all the previous frames and the targeted time reduction. After getting the largest depth of current LCU, it does mode selection from the large depth to the small depth. However, it only decides the maximum depth of each LCU, and has to do mode selections for each depth from the maximum depth to 0. For those LCUs with a lot of detail contents, whose best mode should be with small CU sizes, some redundant computation still remains, since all the LCUs have to search the modes with large CU sizes although they are unlikely the best mode.
4) The algorithm proposed in [18] exploits the correlation of the MV information under different CU sizes according the optical flow of downsampled frames and then determines the CU sizes, which helps to skip some unnecessary depth splitting.
5) The research in [19] presents an algorithm that explores the RD characteristics affected by the symmetric motion partition (SMP) modes and asymmetric motion partition (AMP) modes, and proposes an optimized selection algorithm of SMP and AMP ranges by establishing a function of QP.

Although these researches achieve ignorable performance loss, the time reduction is not significant.

Then, we overview the algorithm aims to save the ME time. Sampaio *et al.* [20] propose to perform ME for each small CU, and then merge the CUs with the same MVs into large PU or CU. However, it is unsuitable for the CUs which belong to the background of the contents and need not perform ME for each small CUs. Besides, in [17], the statistic results show that the computational complexity is increased with the depth increasing. Thus, the ME for small CUs to merge the large CU to be the best mode would cost more time than the ME for large CU directly. Actually, the time reduction of the algorithm proposed in [20] is not obvious.

2) Fast Mode Decision Algorithms Adopted by HEVC Reference Software: Two fast mode decision algorithms [23], [24] have been integrated into the reference software since HM5.0. Gweon *et al.* [23] propose a fast mode decision method to early determine the mode in each depth. In other words, the proposed algorithm helps the encoder skip some PU processes in current depth, however, it still needs to search the following depths. The main idea is based on the condition of the residuals between the reconstructed values and the prediction values: if all the residuals of current CU are zero, all the remaining PU encoding processes of the CU are terminated. Choi *et al.* [24] present a fast mode decision method to early determine whether the subtree computations can be skipped or not, which helps to skip some CU processes in the rest depths. In [24], if the best mode in current depth is SKIP mode, which indicates that the coding mode of the current depth is accurate enough to represent the CU's information, then the following CU division is skipped.

Another fast mode decision algorithm [25] to speed up the encoder computation has been integrated into the reference software since HM6.2. It is in the same spirit with the early skip detection scheme implemented in H.264/AVC [29], [30], but slightly modified to address the different encoding scheme of HEVC. In this way, Inter_2N $\times$ 2N is first checked and then the SKIP mode. All the other modes in the current depth are skipped if the prediction result of current CU satisfies one of the following conditions.

1) All the residuals are zero, and the best mode is Inter_2N $\times$ 2N with zero MVD.
2) All the residuals are zero, and the best mode is merge mode.

TABLE I

PERCENTAGE OF THE TIME USED FOR INTER_2N × 2N AND THE PERCENTAGE OF THE CUs WITH INTER_2N × 2N AS THE BETTER MODE COMPARED WITH MERGE MODE

Size	Sequences	QP	Percentage of time for Inter_2Nx2N RA	LDB	LDP	Inter_2Nx2N better than merge RA	LDB	LDP
1920x1080	pie	20	19.1%	19.8%	15.4%	3.46%	3.57%	3.18%
		30	26.3%	28.7%	24.5%	1.12%	1.72%	2.47%
		40	27.9%	31.0%	27.1%	0.32%	0.38%	0.48%
		avg	24.4%	26.5%	22.3%	1.63%	1.89%	2.04%
	beach	20	18.5%	20.5%	17.7%	0.87%	0.35%	0.50%
		30	22.8%	24.7%	21.4%	1.26%	0.76%	1.14%
		40	26.4%	28.9%	25.5%	0.81%	0.68%	1.08%
		avg	22.6%	24.7%	21.5%	0.98%	0.60%	0.91%
1280x720	crew	20	17.6%	19.3%	16.3%	2.05%	1.56%	1.93%
		30	22.8%	25.2%	22.0%	2.46%	2.45%	3.02%
		40	26.0%	28.9%	25.5%	1.74%	2.24%	2.69%
		avg	22.1%	24.5%	21.3%	2.08%	2.08%	2.55%
	harbour	20	13.7%	15.4%	12.4%	0.87%	0.37%	0.40%
		30	20.5%	21.1%	17.4%	3.58%	4.32%	3.62%
		40	25.1%	26.9%	23.3%	2.19%	2.98%	3.21%
		avg	19.8%	21.1%	17.7%	2.21%	2.56%	2.41%
		avg	22.2%	24.2%	20.7%	1.73%	1.78%	1.98%

Although the algorithms of these proposals are simple and efficient, the computational complexity of HEVC encoder still remains high.

III. PROPOSED FAST MODE DECISION METHOD FOR HEVC

In this section, the proposed fast inter-mode selection algorithm is introduced. First, we analyze the statistical property of the best modes' distribution, and then introduce the method to skip the mode decision of Inter_2N × 2N and the remaining modes including the further CU splitting, which is denoted as Method1. Second, according to the distribution of the distortions in current CU, we propose a fast inter-prediction algorithm to skip parts of the mode selections in current depth, and we denote this algorithm as Method2. Third, we discuss the method to skip the ME operations for PUs and sub-CUs, which is denoted as Method3. At last, we give a description of the whole algorithm's flowchart.

A. Method1: Early SKIP Mode Decision

As introduced in Section II-B, the proposal [25] integrated into the reference software has to check Inter_2N × 2N mode for all the CUs, however, it may be redundant for some CUs. In [25], the encoder first check Inter_2N × 2N for each CU, and then check Merge mode, if the best mode is SKIP mode, then all the remaining modes are skipped. Therefore, in this section, we test the time consuming of Inter_2N × 2N mode and the possibility of choosing Inter_2N×2N as the best mode when SKIP mode is the best mode after checking Merge mode. To make sure of the validity of the proposed algorithm, QP values and test sequences in this section are different to these in the results shown in Section IV. The test condition is as follows: we test some sequences with resolution of 1920 × 1080 and 1280 × 720. The configurations for the encoding are random access, low-delay B (LDB) and LDP with three QPs (20, 30, and 40) being used. As shown in Table I, the time used to check Inter_2N × 2N is more than 20% of the whole encoding time on average, however, less than 2% of the CUs choose Inter_2N × 2N as the best mode if the best mode is SKIP mode after checking merge mode, so we need to find a method to skip checking Inter_2N × 2N mode for the early SKIP mode proposed in [25].

TABLE II

PERCENTAGE OF CUs WITH ZERO MV AS WELL AS SKIP MODE TO THE CUs WITH SKIP MODE

Size	Sequences	QP	percentage of SKIP mode with zero MV RA	LDB	LDP
1920x1080	pie	20	96.2%	93.0%	93.7%
		30	99.4%	98.5%	97.6%
		40	99.8%	99.7%	99.5%
		avg	98.5%	97.1%	96.9%
	beach	20	99.1%	99.2%	98.5%
		30	99.2%	99.3%	98.7%
		40	99.5%	99.7%	99.3%
		avg	99.3%	99.4%	98.8%
1280x720	crew	20	96.0%	93.9%	91.4%
		30	98.5%	97.5%	96.6%
		40	99.3%	98.7%	97.7%
		avg	97.9%	96.7%	95.2%
	harbour	20	96.3%	93.2%	91.9%
		30	94.3%	90.4%	92.2%
		40	98.3%	96.3%	95.3%
		avg	96.3%	93.3%	93.1%
		avg	98.0%	96.6%	96.0%

In this paper, we use the similar idea as early skip method that uses the motion information to early determine SKIP mode as the best mode. We explore the correlation between motion information and Inter_2N × 2N mode to skip Inter_2N × 2N mode selection. As shown in Table II, if the best mode is SKIP mode with zero MV after checking SKIP mode and all the Merge candidates, over 96% of these CUs choose SKIP mode as the best mode after checking all the remaining modes. Thus, in this paper, we first check all

TABLE III
LIST OF THE PARAMETERS

parameters	representation
$NumPartLargeDis$	the number of the parts with distortion greater than $threshold$
$NumPixMoreThres_{idx}$	the number of pixels in part idx with distortion greater than $threshold$
$MaxNumLargeDis$	the maximum value of $NumPixMoreThres_{idx}$ for all parts in current CU

the merge candidates: if the best mode is SKIP mode with zero MV, we determine that the best mode of the current CU is SKIP mode, thus all the other possible modes including Inter_2N × 2N in the current depth and the following depths are skipped. Therefore, compared with the algorithm proposed in [25], there is no need to check Inter_2N × 2N for every CU.

B. Method2: Skip the Mode Decision of Current Depth

Most of the existing methods use the RD-cost of the previous frames to obtain a threshold for current CU's RD-cost, and if the RD-cost is less than the threshold, all the other modes are skipped. However, as discussed in Section II, the RD-costs of the neighboring frames are not similar to each other since HEVC uses the hierarchical coding structure, so it is hard to use the previous frames' RD-costs to predict that of the current frame. Besides, the computation of RD-cost is only taken for each CU instead of each PU. Therefore, we cannot use the RD-cost to decide whether the prediction of current PU is accurate enough and save the encoding time of this PU. In this paper, we propose to use the distribution of the distortions to skip some mode selections in each depth. Different from the existing methods, we define the condition based on the distribution of the distortions in the current CU to decide whether the CU can be split into sub-CUs directly without searching all the modes in current depth. Besides, instead of computing the RD-cost of each PU, we use the distribution of distortions in each PU after Inter_2N × 2N to evaluate whether current depth is suitable for the current CU. For each CU, it is divided into four parts using the quad-tree structure, as shown in Fig. 1, and we use the subparts of the CU to compose the PU of current depth according to the PU splitting method. For example, if the PU splitting method is 2N × N, the top PU consists of the subpart indexed with 0 and 1, and the bottom PU consists of the subpart indexed with 2 and 3.

The distortions of each subpart with different indexes are computed as

$$\mathrm{Dis}_{\mathrm{idx}} = \sum_{i=0}^{H_{\mathrm{sp}}-1} \sum_{j=0}^{W_{\mathrm{sp}}-1} \mathrm{abs}(\mathrm{Rec}_{\mathrm{idx}}[i][j] - \mathrm{Org}_{\mathrm{idx}}[i][j]) \quad (3)$$

where $\mathrm{Dis}_{\mathrm{idx}}$ denotes the distortion of all the pixels in the part with index idx and H_{sp} and W_{sp} are the height and width of the subpart, respectively. $\mathrm{Rec}_{\mathrm{idx}}[i][j]$ and $\mathrm{Org}_{\mathrm{idx}}[i][j]$ are the reconstructed and original pixels in the subpart with index idx whose position is i and j for the row and column, respectively. We define the average distortion of the collocated LCU in the previous picture as

$$\mathrm{Dis}_{\mathrm{avg}} = \frac{\sum_{i=0}^{H_{\mathrm{LCU}}-1} \sum_{j=0}^{W_{\mathrm{LCU}}-1} \mathrm{abs}(\mathrm{Rec}[i][j] - \mathrm{Org}[i][j])}{H_{\mathrm{LCU}} \times W_{\mathrm{LCU}}} \quad (4)$$

where the parameters H_{LCU} and W_{LCU} are the height and width of LCU and $\mathrm{Rec}[i][j]$ and $\mathrm{Org}[i][j]$ are the reconstructed and original pixel values of the nearest reconstructed picture whose position is i and j for the row and column in the collocated LCU, respectively. And we define a parameter $threshold$ which is five times as large as $\mathrm{Dis}_{\mathrm{avg}}$. From the definition of $threshold$, we can see that the distortion larger than $threshold$ is unacceptable since it is five times the average level. Based on the parameter $threshold$, we define some parameters which is listed in Table III to help the encoder analyze the distribution of the distortions. In Fig. 3, the black sub-block means the part having the pixels whose distortion is larger than threshold. And we define the rules as follows.

(a) (b) (c) (d)

Fig. 3. Conditions to skip the mode selections in current depth. (a) All the residuals of the four subblocks are large. (b) Only two diagonal residuals of the subblocks are large. (c) The other two diagonal residuals are large. (d) Only one residual of the four subblocks is large

1) As shown in Fig. 3(a), $NumPartLargeDis$ equals 4, if $MaxNumLargeDis$ is larger than one threshold, then all of the four parts and many of the pixels in each part are with unsatisfied predictions under 2N × 2N mode. In this case, the current depth is inaccurate for the CU. Here, the threshold of $MaxNumLargeDis$ is an experimental value, and it is defined to be 20 for the first depth and 10 for the second depth and 5 for other depth.
2) As shown in Fig. 3(b) and (c), $NumPartLargeDis$ equals 2, and the number of pixels with distortion greater than $threshold$ satisfies (5) or (6), which means that the parts with unacceptable reconstructions are in the diagonal direction. We assume that the current motion information getting from the ME under 2N × 2N granularity is inaccurate for the parts with large distortion, meantime it is accurate enough for the parts with small distortion. Thus, the subparts indexed with 0 and 1 should have different motion information. As a consequence, they should belong to different PU. Therefore, Inter_2N × N, Inter_2N × nU, and Inter_2N × nD can be

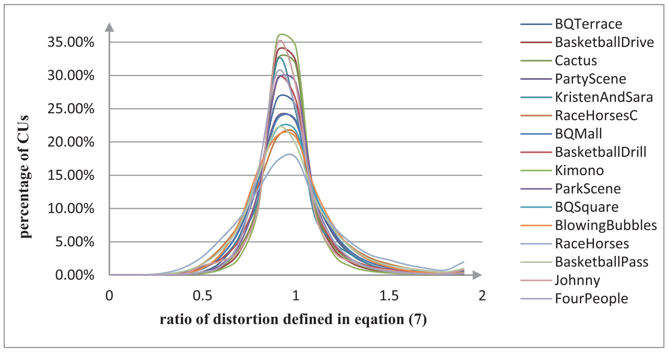


Fig. 4. Distribution of the ratio defined in (7).

skipped. It is similar to the subpart indexed with 0 and 2. In conclusion, the remaining modes in the current depth are not helpful for the cases, so we can split the CU directly

$$\text{NumPixMoreThres}_0 + \text{NumPixMoreThres}_3 = 0 \quad (5)$$
$$\text{NumPixMoreThres}_1 + \text{NumPixMoreThres}_2 = 0. \quad (6)$$

3) As shown in Fig. 3(d), $NumPartLargeDis$ equals 1, which means that only one part of current CU has the prediction which is not good enough. Similar to the explanation to the second rule, the modes in current depth Inter_N × 2N and Inter_2N × N are not helpful for this case, so we can split the CU directly.

Under the rules discussed above, we determine that the current depth is unsuitable for the CU so that we can skip the mode selection in current depth after Inter_2N × 2N and split the CU into sub-CUs directly. Besides, we can skip some of the modes in current depth according to the distribution of the distortions. For example, after Inter_2N × 2N, if the distortion of the left PU roughly equals to the right PU, which indicates that the left PU and the right PU have the similar motion, then there is no need to check Inter_N × 2N. So is the case for Inter_2N × N. Thus, in this paper, we defined the condition as

$$\text{th}_1 < \frac{\text{Dis}_0 + \text{Dis}_2}{\text{Dis}_1 + \text{Dis}_3} < \text{th}_2 \quad (7)$$

when the distortion satisfies (7), we skip checking the mode Inter_N × 2N. And it is similar for Inter_2N × N. Here the parameter th_1 and th_2 are experimental values. Fig. 4 shows the distribution of the ratio defined in (7) for different sequences. From the distribution results, we can see that most of the ratio of the CUs belong to the range of 0.7–1.5. Thus, in all the results provided in this paper, the parameters th_1 and th_2 are set to 0.7 and 1.5, respectively.

C. Method3: Skip Unnecessary ME

Although many fast ME algorithms have been explored to decrease the searching points, the most time-consuming part in HEVC is still ME operation. Therefore, in this section, we first analyze the time spent on ME part. As the result provided in [20], although the fast ME tool (test zone) is turned ON, the time used for ME in average is still more than 70% of the whole encoding time, so finding a method to decrease the ME complexity is necessary. Thus, we propose a method in this paper to skip unnecessary ME of some PUs according to the PUs' residuals. Since ME is the way to find the best prediction for current PU, we define a criterion based on the residual values to determine whether the prediction is good enough for current PU.

After checking Inter_2N × 2N, there has been MV information for current CU with four subparts. Then, we compute the residuals of these subparts as

$$\text{Res}_{\text{idx}} = \sum_{i,j \in \text{part idx}} \text{abs}(\text{Rec}_{\text{idx}}[i][j] - \text{Pred}_{\text{idx}}[i][j]) \quad (8)$$

where Res_{idx} and Pred_{idx} are the residuals and the predictions of the pixels in the part with index idx, and $i, j \in$ part idx means all the pixels in the part with index idx. If the parameter Res_{idx} equals to zero, it means that after inter prediction, transform, and quantization, all the coefficients are zero and there is no bit for residual coding. And it implies that the prediction is good enough for current PU or sub-CU. Therefore, we can copy the motion information and the residuals instead of performing ME for this PU or sub-CU again. For example, when checking Inter_N × 2N mode, if only the residuals of the left PU ($\text{Res}_0 + \text{Res}_2$) are zero, we copy the motion information of current best mode to this PU directly, and only perform ME for the right PU, and vice versa. Besides, when checking the best mode for sub-CUs of current CU, we copy the motion information to the sub-CU with zero residuals directly, and only perform ME for the other sub-CUs whose residuals are not zero. Moreover, if all of the four subparts satisfy the zero residual condition, we keep current best mode as the best mode without checking all the remaining modes.

D. Framework Design

From the above analysis and introduction, we can see that it is possible to improve the fast mode decision in HEVC. First, the computation for Inter_2N × 2N can be saved by some methods. Second, the modes in the same depth can be skipped according to the distribution of the distortions. At last, we can find a way to decrease the ME complexity. In this section, we introduce the flowchart of the proposed fast mode decision method.

Fig. 5 depicts the flow of the proposed algorithm. If the best mode is SKIP mode with zero motion information after checking all the possible merge modes, we can stop searching the remaining modes to save the time used for Inter_2N × 2N. Otherwise, we skip some mode selection operations or ME operations based on the distortions and residuals of each part in the current CU. The detailed algorithm is represented in Algorithm 1, and $skiprange$ is the rules defined in Section III-B.

IV. Experimental Results

To prove the validity of the proposed fast mode decision algorithm, we provide the objective quality results of each proposed method and the overall algorithm. Besides, to verify the effective of the proposed early SKIP method, we provide some subjective results in Section IV-A2. In addition, the

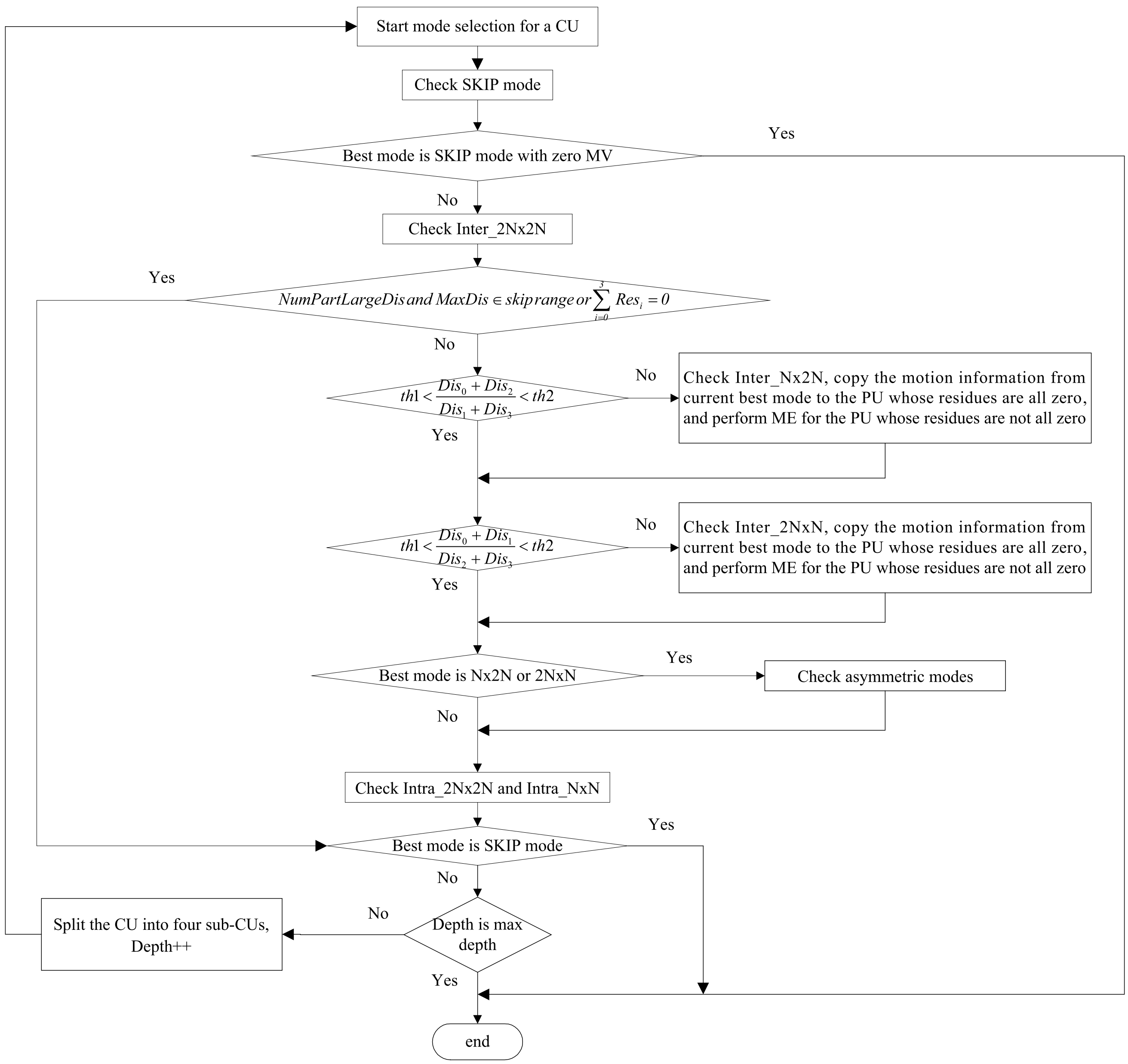


Fig. 5. Flowchart of the proposed overall algorithm.

TABLE IV
TEST PLATFORM

Processor	Intel Core 2 Duo E5649 (2x2.53GHz)
Memory	32GB
Compiler	Microsoft Visual C++ 2010
Operating system	64-bit Microsoft Windows Server 2008 R2

results compared with the existing literatures are presented in Section IV-C. All the results are built with C++ compiler using the function clock() to measure the run time. And the test platform parameters are shown as Table IV.

A. Experimental Results of the Proposed Three Methods

1) Objective Quality Results of All the Methods: As introduced in Section III, in this paper, we propose three methods to speed up the encoder. Method1 aims to early determine SKIP mode as the best mode depending on the MV information. Method2 tries to skip the mode selection operations according to the distribution of the distortions, which splits some CUs into smaller CUs directly without mode selections on large CUs. Method3 attempts to skip the ME operations of PUs or sub-CUs with zero residuals.

To evaluate both the RD performance and the encoding complexity decrease of these three methods, we implement the methods on the HEVC reference software (HM16.4), test all the sequences in Classes A, B, C, and D [31] with different video sizes under random access main profile configuration (RA), and all the sequences in Classes B, C, D, and E [31] under LDP and LDB Main profile configuration. The characteristics of these sequences are summarized in Table V. The proposed algorithm is evaluated with

Algorithm 1 Flowchart of the Proposed Fast Mode Decision Algorithm

Input Parameters:
Threshold, th_1 and th_2; depth and maximal depth, $Depth$ and $Depth_{max}$;
Output Parameters:
The best mode for current CU, $Mode_{best}$;
1: Check all the merge candidates, update $Mode_{best}$: if $Mode_{best} = SKIP$ and $MV_{best} = (0, 0)$, then go to 11; otherwise, go to 2;
2: Check Inter_2N×2N mode, if the residuals of current CU getting from last depth are all zero, copy the motion information from last depth to current CU, update $Mode_{best}$, go to 8; otherwise, perform ME for current CU, and update $Mode_{best}$: if $NumPartLargeDis \in skiprange$ and $MaxDis \in skiprange$, or $\sum_{i=0}^{3} Res_i = 0$, then go to 8; otherwise, go to 3;
3: If $th_1 < \frac{Dis_0+Dis_2}{Dis_1+Dis_3} < th_2$, which means the left PU and the right PU have the similar distortion, then there is no need to test Inter_N×2N, go to 4; otherwise, check Inter_N×2N, copy MV_{best} to the PU whose residuals are zero, perform ME for the PU whose residuals are not zero, and update $Mode_{best}$, then go to 4;
4: If $th_1 < \frac{Dis_0+Dis_1}{Dis_2+Dis_3} < th_2$, which means the top PU and the bottom PU have the similar distortion, then there is no need to test Inter_2N×N, go to 5; otherwise, check Inter_2N×N, copy MV_{best} to the PU whose residuals are zero, perform ME for the PU whose residuals are not zero, and update $Mode_{best}$, then go to 5;
5: If $Mode_{best}$ = Inter_N×2N or Inter_2N×N, go to 6; otherwise, go to 7;
6: Check the asymmetric mode, update $Mode_{best}$, go to 7;
7: Check Intra mode, update $Mode_{best}$, go to 8;
8: If $Mode_{best} = SKIP$, go to 11; otherwise, go to 9;
9: If $Depth = Depth_{max}$, go to 11; otherwise, go to 10;
10: Split the CU into four sub-CUs, $Depth$=$Depth+1$, and then do mode selection for each sub-CU, go to 1;
11: Compare RD-cost of different depth to decide $Mode_{best}$;
12: **return** $Mode_{best}$.

QPs 22, 27, 32, 37, and the other parameters for the test configuration are the same as the CTC [31] of HEVC. The individual evaluation results of these three approaches are shown in Table VI, where the coding efficiency is measured with BD-rate increment [32], and the computational complexity is measured by the amount of time reduction, which is computed as

$$\Delta T = \frac{\text{Time}_{\text{reference}} - \text{Time}_{\text{proposed}}}{\text{Time}_{\text{reference}}} \times 100\%. \tag{9}$$

Table VI shows the BD-rate increment and time reduction comparison between the proposed three methods and HM16.4 anchor under RA test configuration. In Table VI, ΔB means the BD-rate changes (in percentage), and ΔT means the time saving (in percentage). From Table VI, we can see that in average of all the test sequences, Method1 achieves about a 46.0% encoding time saving with only a 0.86% BD-rate loss; Method2 can save 48.3% of encoding time with a 2.74% BD-rate loss; and Method3 can reduce 54.9% of encoding time with a 2.93% BD-rate loss. These results indicate the following.

1) Method1 can efficiently skip unnecessary Inter_2N × 2N mode selection operations and skip the following depths according to the early skip determination.

TABLE V
CHARACTERISTICS OF TEST SEQUENCES

	Number of sequences	Video resolution	Frame rate
Class A	4	2560x1600	30 & 60
Class B	5	1920x1080	24 & 50 & 60
Class C	4	832x480	30 & 50 & 60
Class D	4	416x240	30 & 50 & 60
Class E	3	1280x720	60

2) Method2 can skip the unnecessary PU split mode selection operations in each depth.
3) Method3 can achieve an obvious time reduction by skipping some unnecessary ME operations.

From the purpose of Method1 and Method2, we can see that Method1 aims to early determine SKIP mode as the best mode, which is helpful to skip the further CU splitting. Thus, it is more efficient for the CUs which prefer being coded with large CU sizes. On the contrary, Method2 attempts to split large CUs into small ones directly. Therefore, it is designed for the CUs which tend to be coded with small sizes. To verify this conclusion, we test two sequences of Class A and two sequences of Class B under RA configuration. Table VII shows the time reduction of Method1 and Method2 for different QPs compared with HM16.4 anchor. In sequence *Traffic*, there are many cars driving on the street, many trees and houses are in the background. Although there are many details in the background, they are stationary and can be coded with SKIP mode. Besides, the motion of the cars is translation. Thus, they can be coded with large CU sizes. While in sequence *SteamLocomotive*, the background is filled with details such as trees and snow covered ground. Besides, these details are moving since the camera is zooming. Moreover, the motion is not only translation but also rotation. Therefore, the CUs should be divided into smaller ones to represent these details. Compared with the time reduction of Method1 for these two sequences, we can see that Method1 is more efficient for *Traffic*, while from the results of Method2, it is better for *SteamLocomotive*. The results of these two sequences demonstrate that Method2 is more suitable to represent the context with more details, while Method1 is opposite. As to sequence *Kimono* and *BasketballDrive*, both of these two sequences include the background which can be coded with SKIP mode, but the motion in *BasketballDrive* is more intense. The results of these two sequences indicate that Method2 is more appropriate for the sequences with dramatic motion, while Method1 is opposite. Besides, all the results of these four sequences indicate that Method1 is more efficient for the CUs coded with large QPs, while Method2 is more efficient for the CUs coded with small QPs. As we know, if the sequence is coded with large QP, the encoder prefers choosing large CU sizes. Therefore, the results and the above analysis are coincident. In conclusion, the combination of these two methods is suitable for the CUs with different features under different targeted bit-rate. As for Method3, the results shown in Table VII demonstrate that it provides similar time reduction for the sequences with different motion features.

TABLE VI
RESULTS OF THE PROPOSED METHODS

	sequences	Method1 VS HM16.4 anchor $\Delta B(\%)$	$\Delta T(\%)$	Method2 VS HM16.4 anchor $\Delta B(\%)$	$\Delta T(\%)$	Method3 VS HM16.4 anchor $\Delta B(\%)$	$\Delta T(\%)$
RA	Class A	0.87	50.4	1.83	47.0	2.13	52.8
	Class B	1.21	60.2	1.85	52.8	2.07	56.5
	Class C	1.26	47.2	2.98	46.9	3.45	55.2
	Class D	1.30	46.7	3.00	45.2	3.52	52.6
	Avg	1.16	51.7	2.38	48.3	2.75	54.4
LDP	Class B	0.94	42.9	2.14	48.5	2.21	53.8
	Class C	0.72	29.7	3.10	41.4	3.04	50.6
	Class D	0.71	28.0	3.87	41.0	3.58	48.9
	Class E	0.18	71.7	3.01	52.1	3.37	60.7
	Avg	0.69	41.3	2.98	45.5	2.98	53.1
LDB	Class B	0.73	47.2	2.04	53.1	2.18	54.8
	Class C	0.72	33.4	2.93	48.5	3.07	56.9
	Class D	0.78	31.0	3.77	47.8	3.72	55.8
	Class E	0.55	73.8	2.95	55.3	3.75	64.2
	Avg	0.71	44.7	2.87	51.0	3.08	57.3
	Avg	0.86	46.0	2.74	48.3	2.93	54.9

(a) (b)

Fig. 6. Comparisons of the subjective qualities of the 45th frame of *Kimono* sequence (1920 × 1080). (a) HM16.4, RA (PSNR: 34.03 dB and coding time: 58 s). (b) Method1 (PSNR: 33.87 dB, coding time: 8 s, and speed up of 7.25).

2) Subjective Quality Results of Method1: It has been a well-known characteristics that smartly biasing toward SKIP mode may not have impact on objective quality but most likely have severe dragging artifacts on the picture. Thus, in this section, we provide some subjective quality results of our proposed early SKIP method. We choose the frames with different time reduction and motion activity. In terms of objective results, the peak signal-to-noise ratio (PSNR) of Method1 is slightly lower than that of the HM16.4 anchor. From Figs. 6–8, we can see that, for the sequences with different motion activity and time reduction, there is no significant flaw of image quality for the proposed Method1 from the perspective of subjective quality. In summary, the proposed Method1 and the HM16.4 encoder have similar subjective qualities, but the coding speed of the proposed Method1 is increased significantly.

TABLE VII
TIME REDUCTION RESULTS FOR DIFFERENT QPs

Method	QP	Sequences: Traffic $\Delta T(\%)$	SteamLocomotive $\Delta T(\%)$	Kimono $\Delta T(\%)$	BasketballDrive $\Delta T(\%)$
Method1	22	44.9	28.5	33.0	30.1
	27	61.9	52.3	48.4	44.2
	32	73.1	65.3	62.0	56.2
	37	80.5	74.0	72.2	65.2
	Avg	65.1	55.0	53.9	48.9
Method2	22	46.3	47.8	45.4	48.7
	37	48.8	52.2	47.2	49.5
	32	48.5	50.2	50.3	53.4
	37	47.6	51.9	46.9	51.9
	Avg	47.8	50.5	47.5	50.9
Method3	22	52.8	51.5	52.4	51.7
	37	57.7	57.3	56.7	56.4
	32	61.3	60.2	58.9	60.3
	37	62.5	61.8	61.3	62.9
	Avg	58.6	57.7	57.3	57.8

B. Experimental Results of the Combination Algorithm

In this section, we analyze the experimental result of the proposed combination algorithm which incorporates Method1, Method2, and Method3. To evaluate the performance for sequences with different motion features and to confirm whether the proposed combination algorithm is suitable for different test configurations, we test the proposed combination algorithm for different configurations. All the configurations in HM16.4 specified in [31] except Intra are tested. We test the sequences recommended by Joint Collaborative Team on Video Coding in the highest two resolutions [Classes A and B for RA and Classes B and E for low delay (LD)] covering a wide range of motion activities. The experimental results are presented in Tables VIII–X.

Table VIII shows the BD-rate increment and time reduction of the proposed algorithm compared with HM16.4 anchor and HM16.4 fast anchor (combination of proposals [23]–[25]) under RA test configuration. Tables IX and X show

(a) (b)

Fig. 7. Comparisons of the subjective qualities of the 45th frame of *BasketballDrive* sequence (1920 × 1080). (a) HM16.4, LDP (PSNR: 35.18 dB and coding time: 65 s). (b) Method1 (PSNR: 35.16 dB, coding time: 30 s, and speed up of 2.17).

(a) (b)

Fig. 8. Comparisons of the subjective qualities of the 45th frame of *RaceHorses* sequence (832 × 480). (a) HM16.4, LDB (PSNR: 28.41 dB and coding time: 19 s). (b) Method1 (PSNR: 28.30 dB, coding time: 9 s, and speed up of 2.11).

TABLE VIII

PROPOSED ALGORITHM COMPARED WITH HM16.4 ANCHOR AND HM16.4 FAST ANCHOR IN RA CONFIGURATION

		Proposed VS HM16.4 anchor				Proposed VS HM16.4 fast anchor			
		RA main		RA main10		RA main		RA main10	
size	Sequences	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)
2560	Traffic	4.1	81.3	4.0	81.1	2.5	44.3	2.6	44.4
x1600	Nebuta	1.7	62.3	1.6	62.0	1.2	49.7	1.2	49.3
	SteamLocomotive	2.2	77.3	2.3	77.0	1.9	47.8	2.1	47.6
1920	Kimono	3.0	76.9	3.0	76.8	2.0	48.4	2.0	48.2
x1080	ParkScene	3.7	79.5	3.6	79.3	2.4	45.7	2.3	44.7
	Cactus	5.0	76.1	4.9	75.9	3.1	46.4	3.1	46.8
	BasketballDrive	4.3	74.6	4.2	74.3	2.8	48.9	2.9	48.7
	BQTerrace	4.1	79.2	4.1	79.0	2.7	43.7	2.8	42.5
	Average	3.5	75.9	3.5	75.7	2.3	46.9	2.4	46.5

the results of LD IPPP and low delay B (LDB) structure, respectively. The results reported in these tables indicate that for RA configurations, on average, our proposed algorithm achieves about a 75.8% time reduction with a 3.5% BD-rate increase compared with HM16.4 anchor (with a maximum of 81.3% time reduction in *Traffic* and a minimum 62.3% in *Nebuta*); for LD configurations, a time saving of approximately 78% is achieved with a 4.4% BD-rate increase on average (sequence *Johnny* has a maximum of 86.2% time reduction in LDP and 88.2% in LDB, and sequence *BasketballDrive* has a minimum 68.5% in LDP and 74.3% in LDB). In summary, for all the test configurations, on average, the proposed method achieves about a 77% time reduction while the BD-rate increase is only 4.1% compared with HM16.4 anchor, and achieves about a 48% time reduction with only a 2.9% extra BD-rate compared with HM16.4 fast anchor. The results demonstrate that the proposed combination algorithm is suitable for all the sequences with different motion activities under different test configurations, and the time reduction is more obvious for the sequences with low motion activities.

C. Performance Compared With the State-of-the-Art Fast Algorithm

1) Performance Compared With the State-of-the-Art Fast Algorithm With Similar Time Reduction: We compare Method1 with the state-of-the-art fast algorithms proposed in [16] and [18]. Since [16] only provides the results under

TABLE IX
PROPOSED ALGORITHM COMPARED WITH HM16.4 ANCHOR AND HM16.4 FAST ANCHOR IN LDP CONFIGURATION

size	Sequences	Proposed VS HM16.4 anchor				Proposed VS HM16.4 fast anchor			
		LDP main		LDP main10		LDP main		LDP main10	
		ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)
1920	Kimono	2.8	70.3	2.9	70.0	2.0	49.0	2.0	48.8
x1080	ParkScene	4.8	72.5	4.8	72.3	3.6	47.7	3.6	46.7
	Cactus	5.3	70.6	5.2	70.2	3.4	47.7	3.4	47.8
	BasketballDrive	3.5	68.5	3.5	68.7	2.3	48.2	2.3	48.8
	BQTerrace	5.2	72.8	5.2	72.5	3.9	45.5	4.0	45.9
1280	FourPeople	4.3	82.6	4.1	82.2	3.4	43.0	3.3	44.5
x720	Johnny	4.9	86.2	4.7	86.0	3.7	41.2	3.6	40.8
	KristenAndSara	4.2	83.5	4.3	83.1	3.6	44.7	3.5	44.8
	Average	4.4	75.9	4.3	75.6	3.2	45.9	3.2	46.0

TABLE X
PROPOSED ALGORITHM COMPARED WITH HM16.4 ANCHOR AND HM16.4 FAST ANCHOR IN LDB CONFIGURATION

size	Sequences	Proposed VS HM16.4 anchor				Proposed VS HM16.4 fast anchor			
		LDB main		LDB main10		LDB main		LDB main10	
		ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)
1920	Kimono	3.3	75.6	3.4	74.5	2.4	54.5	2.5	54.3
x1080	ParkScene	4.4	77.2	4.5	77.0	3.5	53.0	3.4	53.1
	Cactus	4.8	74.4	4.7	74.0	3.3	52.3	3.3	52.3
	BasketballDrive	3.8	74.3	3.8	73.8	2.8	53.8	2.8	53.9
	BQTerrace	3.7	78.7	3.9	77.4	3.1	49.9	3.0	50.2
1280	FourPeople	4.8	85.1	4.7	84.9	3.5	48.5	3.0	48.2
x720	Johnny	5.9	88.2	5.7	88.0	4.2	46.3	4.0	45.7
	KristenAndSara	5.0	85.8	4.8	85.5	3.8	48.5	3.5	48.9
	Average	4.5	79.9	4.4	79.4	3.3	50.9	3.2	50.8

TABLE XI
PROPOSED METHOD1 COMPARED WITH THE ALGORITHMS PROPOSED IN [16]

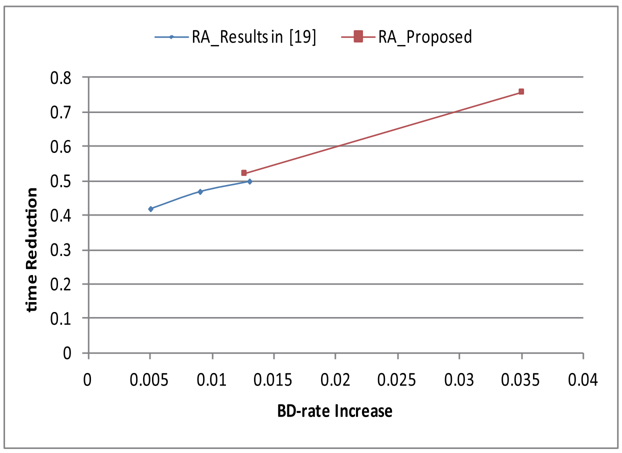
size	Sequences	RA				LDP			
		[16]		Method1		[16]		Method1	
		ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)
ClassA	Traffic	1.38	40	1.34	68.6	–	–	–	–
	PeopleOnStreet	1.30	25	1.25	37.9	–	–	–	–
ClassB	Kimono	0.53	39	0.91	58.5	0.40	36	0.68	40.0
	ParkScene	1.24	36	1.14	65.0	0.95	33	0.89	47.0
	Cactus	1.02	36	1.76	58.3	0.84	34	1.85	41.1
	BasketballDrive	1.25	37	1.00	54.0	1.11	36	0.71	37.3
	BQTerrace	0.86	37	1.26	65.2	0.87	35	0.55	49.1
ClassC	RaceHorsesC	1.68	21	1.41	36.3	1.10	20	0.65	20.6
	BQMall	2.74	29	1.45	56.3	2.13	26	0.79	38.0
	PartyScene	0.66	25	1.23	48.9	0.27	21	0.84	29.0
	BasketballDrill	1.16	31	0.93	47.3	1.40	31	0.61	31.1
ClassD	RaceHorses	0.70	14	1.75	34.1	0.51	14	0.90	18.2
	BQSquare	0.85	20	0.73	60.2	1.11	16	0.14	37.8
	BlowingBubbles	0.82	19	1.08	51.3	0.69	16	0.89	31.2
	BasketballPass	1.57	19	1.65	41.0	1.93	19	0.89	24.8
	Average	1.18	28.5	1.26	52.2	1.02	25.9	0.80	34.2

RA and LDP configuration, while [18] only provides the results under LDP and LDB configuration, we compared with these two algorithms by providing the results under the same configurations with the corresponding algorithms. Table XI shows the results under RA and LDP configurations compared with the algorithm proposed in [16], and Table XII shows the results under LDP and LDB configurations compared with the algorithm proposed in [18]. We can see from Table XI that the proposed Method1 achieves obviously more time reduction with similar BD-rate increase compared with [16]. From Table XII, we can see that Method1 achieves similar time reduction with only about half BD-rate increase compared with [18].

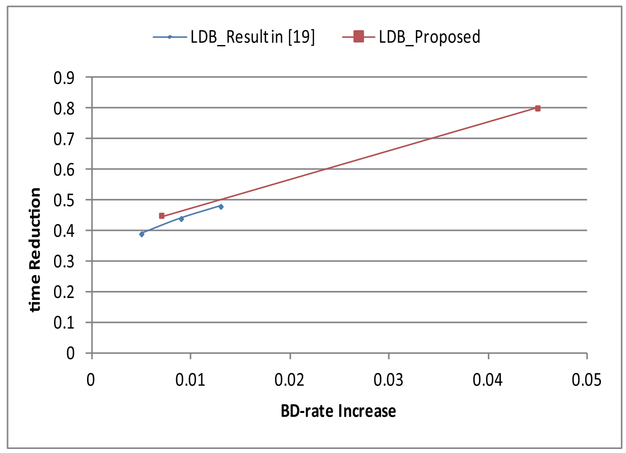
2) Performance Compared Combined Algorithm With the State-of-the-Art Fast Algorithm: In most of the existing fast algorithms designed for HEVC, the time reduction is about 50%. However, the combined algorithm in this paper provides a significant time reduction. Thus, it is hard to compare the combined algorithm with the existing fast algorithm directly. Therefore, we choose [19] as the benchmark to compare with combined algorithm, since it provides the results covered wide time reduction with wide BD-rate increase range and

TABLE XII
PROPOSED METHOD1 COMPARED WITH THE ALGORITHMS PROPOSED IN [18]

		LDB				LDP			
		[18]		Method1		[18]		Method1	
size	Sequences	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)	ΔB (%)	ΔT (%)
ClassB	Kimono	2.36	36.7	0.73	43.9	1.61	47.0	0.68	40.0
	ParkScene	2.53	39.8	0.79	50.7	2.01	42.7	0.89	47.0
	Cactus	2.48	43.8	1.35	45.7	2.14	42.2	1.85	41.1
	BasketballDrive	4.60	46.0	0.74	41.7	3.88	51.9	0.71	37.3
	BQTerrace	1.62	31.9	0.06	54.0	0.60	51.5	0.55	49.1
ClassC	RaceHorsesC	1.99	30.9	0.60	23.6	2.62	31.8	0.65	20.6
	BQMall	1.98	42.4	0.88	42.4	3.16	47.3	0.79	38.0
	PartyScene	1.54	30.3	0.68	32.4	1.06	32.3	0.84	29.0
	BasketballDrill	1.73	44.2	0.70	35.0	1.74	34.7	0.61	31.1
ClassD	RaceHorses	1.54	31.1	0.90	20.8	1.19	31.2	0.90	18.2
	BQSquare	2.35	33.0	0.53	40.2	1.14	24.2	0.14	37.8
	BlowingBubbles	1.68	25.6	0.78	34.2	2.12	32.1	0.89	31.2
	BasketballPass	1.17	44.5	0.90	28.7	1.81	41.1	0.89	24.8
	Average	2.12	36.9	0.74	37.9	1.93	39.2	0.80	34.2



(a)



(b)

Fig. 9. Comparisons of the objective quality and time reduction with the state-of-the-art fast algorithm [19]. (a) Result under RA configuration. (b) Result under LDB configuration.

the results of our proposed methods have some overlap with the results in [19]. Fig. 9 shows the time savings and BD-rate increase provided by the algorithm proposed in [19] with different parameter settings and the proposed algorithms in this paper. The blue dots stand for the performance of the algorithm proposed in [19] with various parameters to obtain different time reduction, and the red dots stand for the performance of Method1 and combined algorithm. From Fig.9, it is obvious that the proposed algorithm has better time reduction compared with [19].

V. CONCLUSION

The HEVC standard employs a hybrid coding scheme similar to the H.264/AVC standard. However, different from H.264/AVC, the hierarchical coding structure and the flexible CU sizes make the fast mode decision more difficult. In this paper, we propose a fast mode selection algorithm based on the distribution of the distortions and the residuals, which helps to skip some modes in current depth and the ME for some PUs. Some extensive experiments are performed on video sequences with different characteristics under different test configurations. The experimental results show that the proposed method can save about 77% of encoding time with only a 4.1% BD-rate increase on average compared with HM16.4 anchor, and compared with the fast mode decision method adopted in HM16.4, the proposed algorithm achieves about a 48% encoding time reduction with only a 2.9% BD-rate increase. From the results, we can see that the proposed fast mode decision algorithm can speed up the encoder significantly with acceptable BD-rate increase, and it is suitable for low-delay applications.

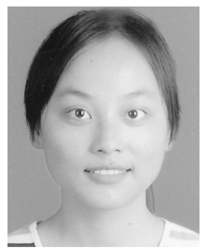
REFERENCES

[1] 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.

[2] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[4] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 207–213, Feb. 2013.
[5] 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.
[6] H.-Y. C. Tourapis and A. M. Tourapis, "Fast motion estimation within the H.264 codec," in *Proc. Int. Conf. Multimedia Expo (ICME)*, vol. 3. 2003, pp. III-517–III-520.
[7] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.
[8] J. Ren, N. Kehtarnavaz, and M. Budagavi, "Computationally efficient mode selection in H.264/AVC video coding," *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 877–886, May 2008.
[9] J. Y. Lee and H. Park, "A fast mode decision method based on motion cost and intra prediction cost for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 393–402, Mar. 2012.
[10] S.-H. Ri, Y. Vatis, and J. Ostermann, "Fast inter-mode decision in an H.264/AVC encoder using mode and Lagrangian cost correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 302–306, Feb. 2009.
[11] Y.-H. Sung and J.-C. Wang, "Fast mode decision for H.264/AVC based on rate-distortion clustering," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 693–702, Jun. 2012.
[12] E. Martínez-Enríquez, A. Jiménez-Moreno, and F. Diaz-de-Maria, "An adaptive algorithm for fast inter mode decision in the H.264/AVC video coding standard," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 826–834, May 2010.
[13] B.-G. Kim and C.-S. Cho, "A fast inter-mode decision algorithm based on macro-block tracking for *P* slices in the H.264/AVC video standard," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 5. Sep./Oct. 2007, pp. V-301–V-304.
[14] B.-G. Kim, "Novel inter-mode decision algorithm based on macroblock (MB) tracking for the P-slice in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 273–279, Feb. 2008.
[15] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 447–452, Mar. 2006.
[16] 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.
[17] G. Correa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.
[18] 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.
[19] 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.
[20] F. Sampaio, S. Bampi, M. Grellert, L. Agostini, and J. Mattos, "Motion vectors merging: Low complexity prediction unit decision heuristic for the inter-prediction of HEVC encoders," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2012, pp. 657–662.
[21] 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.
[22] 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.
[23] R.-H. Gweon and Y.-L. Lee, *Early Termination of CU Encoding to Reduce HEVC Complexity*, document JCTVC-F045, JCT-VC, Turin, Italy, Jul. 2011.
[24] K. Choi, S. H. Park, and E. S. Jang, *Coding Tree Pruning Based CU Early Termination*, document JCTVC-F092, JCT-VC, Turin, Italy, Jul. 2011.
[25] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, *Early SKIP Detection for HEVC*, document JCTVC-G543, JCT-VC, Geneva, Switzerland, Nov. 2011.
[26] C. Lim, S. M. T. Naing, V. Wahadanish, and X. Jing, *Reference Lists for B Pictures Under Low Delay Constraints*, document JCTVC-D093, JCT-VC, Daegu, Korea, Jan. 2011.
[27] H. Li, B. Li, and J. Xu, "Rate-distortion optimized reference picture management for High Efficiency Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1844–1857, Dec. 2012.
[28] C. E. Rhee, K. Lee, T. S. Kim, and H.-J. Lee, "A survey of fast mode decision algorithms for inter-prediction and their applications to High Efficiency Video Coding," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1375–1383, Nov. 2012.
[29] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557–1561, Dec. 2006.
[30] B. W. Jeon and J. Y. Lee, *Fast Mode Decision for H.264*, document JVT-J033, JVT, Waikoloa, HI, USA, Dec. 2003.
[31] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-L1100, JCT-VC, Geneva, Switzerland, Jan. 2013.
[32] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, ITU-T SG16 Q.6, Austin, TX, USA, Apr. 2001.



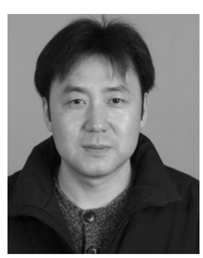
Jinlei Zhang received the B.S. degree in electronics engineering from the University of Science and Technology of China, Hefei, China, in 2010, where she is currently working toward the Ph.D. degree with the Department of Electronic Engineering and Information Science.

Her research interests include video coding, processing, and image compression.

Bin Li (M'14) received the B.S. and Ph.D. degrees in electronics engineering from University of Science and Technology of China, Hefei, China, in 2008 and 2013, respectively.

He joined Microsoft Research Asia, Beijing, China, as an Associate Researcher, in 2013. He holds more than 30 technical proposals that have been adopted by the Joint Collaborative Team on Video Coding. He has authored or co-authored over ten papers. He holds over ten granted or pending U.S. patents in image and video coding. His research interests include video coding, processing, and communication.

Dr. Li received the best paper award of the International Conference on Mobile and Ubiquitous Multimedia from the Association for Computing Machinery in 2011. He was a recipient of the Top 10% Paper Award of the IEEE International Conference on Image Processing in 2014. He has been an Active Contributor to ISO/MPEG and ITU-T video coding standards with the Joint Collaborative Team on Video Coding. He is currently the Co-Chair of the Ad Hoc Group of Screen Content Coding Extensions Software Development.



Houqiang Li (S'12–SM'12) received the B.S., M.Eng., and Ph.D. degrees in electronics engineering from University of Science and Technology of China (USTC), Hefei, China, in 1992, 1997, and 2000, respectively.

He is a Professor with the Department of Electronic Engineering and Information Science, USTC. He has authored or co-authored over 100 papers in journals and conferences. His research interests include video coding and communication, image/video analysis, and multimedia search.

Dr. Li was a recipient of the best paper award of Visual Communications and Image Processing in 2012, the International Conference on Internet Multimedia Computing and Service in 2012, and the International Conference on Mobile and Ubiquitous Multimedia from ACM MUM in 2011, and a senior author of the Best Student Paper of the Fifth International Mobile Multimedia Communications Conference in 2009. He served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2013, and has been on the Editorial Board of *Journal of Multimedia* since 2009.