Fast Motion Estimation Based on Content Property for Low-Complexity H.265/HEVC Encoder

Zhaoqing Pan, *Member, IEEE*, Jianjun Lei, *Member, IEEE*, Yun Zhang, *Senior Member, IEEE*, Xingming Sun, *Senior Member, IEEE*, and Sam Kwong, *Fellow, IEEE*

***Abstract*—The high definition (HD) and ultra HD videos can be widely applied in broadcasting applications. However, with the increased resolution of video, the volume of the raw HD visual information data increases significantly, which becomes a challenge for storage, processing, and transmitting the HD visual data. The state-of-the-art video compression standard-H.265/High Efficiency Video Coding (HEVC) compresses the raw HD visual data efficiently, while the high compression rate comes at the cost of heavy computation load. Hence, reducing the encoding complexity becomes vital for the H.265/HEVC encoder to be used in broadcasting applications. In this paper, based on the best motion vector selection correlation among the different size prediction modes, we propose a fast motion estimation (ME) method to reduce the encoding complexity of the H.265/HEVC encoder. First, according to the prediction unit (PU) partition type, all PUs are classified into two classes, parent PU and children PUs, respectively. Then, based on the best motion vector selection correlation between the parent PU and children PUs, the block matching search process of the children PUs is adaptively skipped if their parent PU chooses the initial search point as its final optimal motion vector in the ME process. Experimental results show that the proposed method achieves an average of 20% ME time saving as compared with the original HM-TZSearch. Meanwhile, the rate distortion performance degradation is negligible.**

***Index Terms*—Video compression, H.265/HEVC, fast motion estimation, prediction unit.**

Manuscript received November 2, 2015; revised May 10, 2016; accepted June 6, 2016. Date of publication June 28, 2016; date of current version August 31, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61501246, Grant 61271324, Grant 61471348, and Grant 61232016, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20150930, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 15KJB510019, in part by the Natural Science Foundation of Hebei Province of China under Grant F2015202311, in part by the Project through the Priority Academic Program Development of Jiangsu Higher Education Institutions, in part by the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology under Grant 2015r012; and in part by the Natural Science Foundation of Guangdong Province for Distinguished Young Scholar under Grant 2016A030306022.
(Corresponding author: Jianjun Lei.)

Z. Pan is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China, also with the Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the School of Computer Science and Engineering, Hebei University of Technology, Tianjin 300401, China (e-mail: zqpan3-c@my.cityu.edu.hk).

J. Lei is with the School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: jjlei@tju.edu.cn).

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

X. Sun is with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: xmsun2013@gmail.com).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cssamk@cityu.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/TBC.2016.2580920

I. Introduction

THE HIGH definition (HD) and ultra HD videos have emerged in response to the developments in video capture and display technologies, which have been widely used in security surveillance, ultra HD television system, and so on [1]. However, with the increased resolution of video, the volume of the raw HD visual information data increases significantly. This becomes a challenge for storage, processing and transmitting the visual data due to the current storage, computing, and transmission capability are still limited. Hence, the high compression rate and low complexity are the key requirements for the HD videos to be widely applied in broadcasting applications. Recently, the joint collaborative team on video coding (JCT-VC) under the ITU-T video coding experts group (VCEG) and ISO/IEC moving picture experts group (MPEG) launched a state-of-the-art video compression standard called H.265/high efficiency video coding (HEVC) [2], [3]. Compared to the H.264/advanced video coding (AVC) which is the previous generation video compression standard [4], the H.265/HEVC achieves about 50% bit rate saving while maintaining the same subjective visual quality. However, the higher compression rate comes at the cost of heavy computational complexity of a series of advanced coding tools used in the H.265/HEVC encoder, such as quadtree structure based coding unit (CU), large and asymmetric interframe/intraframe prediction unit (PU), and so on. Hence, reducing the encoding complexity becomes vital for the H.265/HEVC encoder to be widely used in broadcasting applications.

In order to reduce the encoding complexity of the H.265/HEVC encoder, many researchers have focused on optimizing the H.265/HEVC interframe prediction process [5]–[10]. Pan *et al.* [5] proposed a fast CU size decision method by using the CU quadtree depth selection correlation between the current CU and its spatial and temporal neighboring CUs. Since the CU size highly depends on the content of the CU, Shen *et al.* [6] proposed a fast CU size decision by using the correlation between the CU size and the CU content. Zhang *et al.* [7] proposed a machine learning based fast CU size decision method, which optimized the encoding complexity with acceptable rate distortion (RD) performance degradation. Based on the CU motion activity and hierarchial depth correlation, Pan *et al.* [8] proposed an early Merge/Skip mode decision for reducing the encoding complexity of PU encoding process. By using the CU inter-level and spatiotemporal correlations, Shen *et al.* [9] proposed an adaptive inter mode decision for reducing the complexity of the H.265/HEVC. Based on the spatiotemporal encoding parameters of the H.265/HEVC encoder, Ahn *et al.* [10] proposed a fast CU encoding method for H.265/HEVC inter coding. These methods can efficiently reduce the computational complexity of the interframe prediction process, however, the encoding complexity of the interframe prediction mainly comes from the motion estimation (ME) process, the encoding complexity of H.265/HEVC encoder can be further reduced by optimizing the ME process.

To further reduce the computational complexity of the ME process, many fast ME algorithms have been proposed, such as the conventional fast ME algorithms [11]–[14], including three step search [11], four step search [12], diamond search [13], hexagon search [14], and so on. Bossen *et al.* [15] proposed a hybrid unsymmetrical-cross multi-hexagon-grid search algorithm, which efficiently solves the local minimum problem in ME process. In [16], according to the best motion vectors (MVs) distribution, Pan *et al.* [16] proposed a fast ME algorithm for reducing the computational complexity of the ME process of the H.264/AVC encoder. In [17], an efficient ME and disparity estimation (DE) was proposed for H.264/AVC based multiview video coding by considering the early SKIP mode decision, ME/DE early termination, and adaptive ME/DE search range reduction. However, these fast ME algorithms are not suitable for directly applying to the H.265/HEVC encoder due to the large resolution videos, and new video coding techniques such as quadtree based CU, larger and asymmetrical PU.

In order to address the limitations and full use the properties of the new coding techniques, researchers have devoted their efforts on designing fast ME algorithms for the H.265/HEVC [18]–[24]. In [18], a fast ME algorithm was proposed by refining the search pattern and adopting an early termination for the TZSearch in the H.265/HEVC reference software. By using the hexagon search instead of the TZSearch, a fast ME method was developed for the H.265/HEVC [19]. Based on the content property of different-size CUs and best MVs distribution, an early termination was proposed for the TZSearch in the H.265/HEVC reference software [20]. In [21], a fast ME algorithm was proposed by using the block motion intensity which is evaluated by the MV and MV differences of nearby blocks, in which if the motion intensity is large, the original TZSearch is used; otherwise, the hexagon search is adopted. By considering the highly probable movement in horizontal and vertical direction, Yang *et al.* [22] proposed a directional search based fast ME. In [23], a search window selection method was designed for the HEVC fast ME, in which the search window is determined according to the MV predictor of the size of 64×64 CU. In [24], an adaptive search range decision method was proposed for the HEVC ME, in which the relationships among

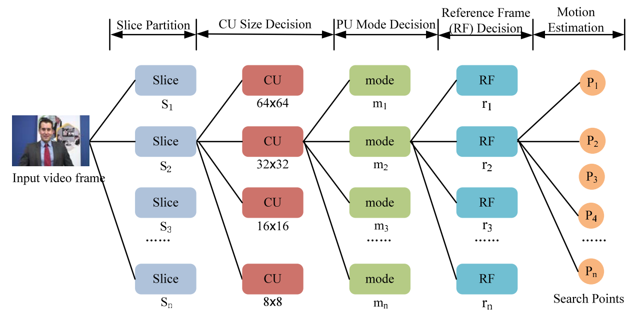


Fig. 1. Encoding steps of the H.265/HEVC interprediction.

the PU difference, predictive MV length, and the best search range are firstly learned; then the k-nearest neighbor algorithm is applied to determine the optimal search range based on the learning results. While the encoding complexity of ME can be further improved due to the block matching search and early termination strategy still cost extra computational complexity. Moreover, the best MV decision correlation among different PU modes are not considered by these algorithms.

In this paper, we propose a content property based fast ME method for reducing the computational complexity of the H.265/HEVC encoder. The rest of this paper is organized as follows. Section II presents a brief review on the ME process of the H.265/HEVC. Section III introduces the details of the proposed content similarity based fast ME method. Experimental results are given in Section IV. At last, Section V concludes this paper.

II. Review on the H.265/HEVC ME Encoding Process

In the H.265/HEVC encoding process, all images/frames of the video are encoded sequentially. As shown in Fig. 1, each frame is partitioned into one or more slice ($S_1, S_2, \cdots, S_n$), the slices are further split into a series of coding tree units (CTUs), which is the basic processing unit of the H.265/HEVC encoder. The CTUs are further partitioned into a sequence of CUs according to a quadtree, which supports the CU size from 64×64 to 8×8. After that, based on the prediction type, a CU can be split into one, two, or four PUs, which is the basic processing unit of intra and inter prediction. For the H.265/HEVC inter prediction, eight inter PU modes are supported, i.e., Inter_2N×2N, Inter_2N×N, Inter_N×2N, Inter_N×N, Inter_2N×nU, Inter_2N×nD, Inter_nL×2N, and Inter_nR×2N. In order to remove the temporal redundancies, these inter PU modes perform the ME in the reference frames (i.e., $r_1, r_2, \cdots, r_n$.) to locate the best matching block according to the minimization of the Lagrangian RD cost function [25].

The H.265/HEVC fast integer-pixel ME process consists of two steps, i.e., initial search point (ISP) decision, and block matching search. For the ISP decision process, the median predictor [15] is used as one of the ISP candidates to predict the ISP, the MV obtained by the median predictor, $\vec{MV}_{MP}$, is shown as

$$\vec{MV}_{MP} = \text{Median}\left(\vec{MV}_{left}, \vec{MV}_{up}, \vec{MV}_{up\text{-}right}\right), \tag{1}$$

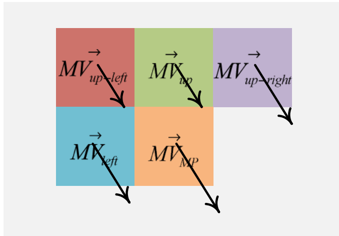


Fig. 2. An example of the nearby blocks of the current block.

it uses the median MV value of the left, up, and up-right blocks of the current block as the ISP. Fig. 2 shows the nearby blocks on the left, up, and up-right of the current CU, and their MVs are denoted as $\vec{MV}_{left}$, $\vec{MV}_{up}$, $\vec{MV}_{up\text{-}right}$, respectively. The other ISP candidate is the $(\vec{0,0})$. Finally, the best ISP, $\vec{MV}_{ISP}$, is determined according to the minimization of the Lagrangian RD cost function, that is

$$\begin{cases} \vec{MV}_{ISP} = \arg\min_{\vec{MV}_n} J(\vec{MV}_n, \lambda_{MOTION}), \ s.t.\ \vec{MV}_n \in S, \\ J(\vec{MV}_n, \lambda_{MOTION}) = SAD(o, c(\vec{MV}_n)) \\ \qquad + \lambda_{MOTION} \cdot R(\vec{MV}_n - \vec{PMV}), \\ SAD(o, c(\vec{MV})) \\ \qquad = \sum_{x=1}^{p_x} \sum_{y=1}^{p_y} \left| o(x, y) - c(x - MV_x, y - MV_y) \right|, \end{cases} \tag{2}$$

where $\vec{MV}_n$ is the MV of the candidate ISP; $S = \{\vec{MV}_{MP}, (\vec{0,0})\}$; SAD is the sum of absolute difference between the original PU (o), and its predicted PU (c), at the position located by $\vec{MV}_n$ in the reference frame; λ_{MOTION} is the Lagrange multiplier; $\vec{PMV}$ is the prediction vector; $R(\vec{MV}_n - \vec{PMV})$ represents the number of bits for encoding the motion information, which is obtained by a table-lookup; p_x, p_y is the PU size, and is equal to 64, 32, 16, or 8.

After the best ISP is determined, a search window with search range w is constructed on the reference frame. Next, a block matching search is performed on the search window to locate the best search point with MV, $\vec{MV}_{best}$, according to the minimization of the Lagrangian RD cost function, which is shown in Eq. (2),

$$\vec{MV}_{best} = \arg\min_{\vec{MV}_n} J(\vec{MV}_c, \lambda_{MOTION}), \ s.t.\ \vec{MV}_c \in (\vec{MV}_m \cup \vec{MV}_{ISP}), \tag{3}$$

where $\vec{MV}_c$ indicates the MVs of all candidate search points; $\vec{MV}_m$ means the MVs of candidate search points in the search window; $\vec{MV}_{ISP}$ represents the MV of ISP. Theoretically, the computational complexity of the block matching search based ME depends on the search window size and the search strategy, the number of candidate search points of the block matching search increases as the size of search window and the complexity of the search strategy increase. Thus, if the block matching search is simplified or skipped, the number of candidate search points will be reduced significantly, which will result in much more computational complexity saving.

TABLE I
TEST CONDITIONS

Max. CU size	64×64
Max. CU depth	4
Quantization Parameter (QP)	22, 27, 32, 37
ME Method	TZSearch
Search Range	[-64,64]
Coding Profile	Low-Delay-Main (LDM) Random-Access-Main (RAM)

III. PROPOSED CONTENT PROPERTY BASED LOW-COMPLEXITY H.265/HEVC ME

A. Encoding Complexity Analyses on the H.265/HEVC ME

The encoding complexity of H.265/HEVC ME depends on the total number of search points of block matching process, suppose a CU is with size of 2N×2N pixels and the ME search range is $\pm w$ in both horizontal and vertical directions, there are $(2w + 1)^2$ candidate search points inside the search window. Hence, if the number of search points is reduced, the computational complexity of the ME could be saved. The TZSearch is a fast ME method in the H.265/HEVC reference software, it reduces the number of search points of the ME by simplifying the block matching search process. However, with these advanced coding tools used, such as quadtree structure based CU, larger and asymmetrical PU modes, the encoding complexity of the TZSearch based fast ME is still high. To analyze the encoding complexity of the ME process in the H.265/HEVC encoder, the H.265/HEVC reference software HM12.0 is used. Five HEVC standard video sequences ("BQSquare", "BasketballDrill", "Johnny", "Cactus", "Traffic") are tested, the property of these five test sequences and HM12.0 configurations are defined in [26], and the resolutions of these five sequences are 416×240, 832×480, 1280×720, 1920×1080, and 2560×1600, respectively. The number of encoded frames for these five sequences is 161, 129, 97, 65, and 33, respectively. Among these video sequences, the "BQSquare" is with medium motion; the "BasketballDrill" moves fast; the "Johnny" has slow motion, and simple background; the "Cactus" is with complex background, and the objects move fast; the "Traffic" has complex content, and the objects have medium motion. The test conditions are tabulated in Table I. The other coding parameters adopts the default settings in the HM12.0 and coding profiles. In this paper, the encoding complexity of the ME, $ME_{complexity}$, is computed as

$$ME_{complexity} = (T_{ME}/T_{encoder}) \times 100\%, \tag{4}$$

where the T_{ME} indicates the encoding time consumed by the ME process; $T_{encoder}$ means the total encoding time of the HM12.0. The statistical results are listed in Table II.

From Table II, it can be observed that the encoding complexity of the variable-sizes CU and PU based ME is quite time-consuming, for the random-access-main (RAM) coding profile, the $ME_{complexity}$ is from 51.39% to 75.33%, 67.82% on average; for the low-delay-main (LDM) coding profile, the $ME_{complexity}$ is from 55.05% to 83.15%, 73.78% on average. In addition, we can see that with the increased value of quantization parameter (QP), the value of $ME_{complexity}$ becomes large, this is because that when the QP increases, the total encoding

TABLE II
STATISTICAL RESULTS OF THE H.265/HEVC ME ENCODING COMPLEXITY (%)

Profile	Sequence	QP=22	QP=27	QP=32	QP37	Average
RAM	BQSquare	51.39	62.77	69.47	73.72	64.34
	BasketballDrill	59.97	65.97	71.54	74.61	68.02
	Johnny	n/a[1]	n/a	n/a	n/a	n/a
	Cactus	58.80	68.45	72.70	74.75	68.68
	Traffic	62.55	69.44	73.61	75.33	70.23
	Average	58.18	66.66	71.83	74.60	67.82
LDM	BQSquare	55.05	66.50	73.70	78.84	68.52
	BasketballDrill	64.77	71.30	76.55	79.57	73.05
	Johnny	74.23	80.03	82.44	83.15	79.96
	Cactus	63.04	72.94	77.97	80.42	73.59
	Traffic	n/a	n/a	n/a	n/a	n/a
	Average	64.27	72.69	77.67	80.50	73.78

time decreases, while the change of ME time is rather small. We can also see that for different video sequences, the values of $ME_{complexity}$ are different, this is because the motion activity and texture of the video content are different. Usually, the video content with complex texture and violent motion activity needs more ME encoding time, instead, the value of $ME_{complexity}$ is small. From these values, we can figure out that the encoding complexity of ME of H.265/HEVC encoder is high, which leaves a lot of room for optimizing the encoding complexity. Thus, if the block matching search process of the ME can be skipped, much more encoding time could be saved.

B. Proposed Content Similarity Based Fast ME Method

1) Initial Search Point Decision: In video coding process, there is large MV correlation among the current CU and its spatial neighboring CUs, such as up, left, and up-right CUs, due to the consistency of object. The block matching search ME methods are based on the assumption that the block matching error decreases monotonously when approaching the global optimal search point. Hence, a more accurate ISP will result in low encoding complexity in finding the global optimal search point. In this paper, the median predictor ($\vec{MV}_{MP}$) and $(\vec{0,0})$ are also adopted as the ISP candidates. Ultimately, the ISP, $\vec{MV}_{ISP}$, is that

$$\vec{MV}_{ISP} = \arg\min_{\vec{MV}_n} J\left(\vec{MV}_n, \lambda_{MOTION}\right), s.t.\ \vec{MV}_n \in \left\{\vec{MV}_{MP}, (\vec{0,0})\right\}. \tag{5}$$

In order to evaluate the efficiency of the ISP, let the event **A** represents that the final optimal search point is same with the ISP in ME process. The probability $P(\mathbf{A})$ is analyzed with five HEVC test video sequences, the test conditions are listed in Table I. The statistical results of $P(\mathbf{A})$ are tabulated in Table III. From Table III, it can be seen that most of the PUs select the ISP as their final global optimal search point in the ME process. For the RAM coding profile, from 61.06% to 75.26%, 69.27% on average, PUs select the ISP as their final global search point. For the LDM coding profile, from 55.90% to 85.30%, 68.01% on average, PUs choose the ISP as their final

[1]The "n/a" indicates the result is not available, since the sequence "Johnny" with the RAM coding profile, and the sequence "Traffic" with the LDM coding profile are not tested according to the HEVC common test conditions [26].

TABLE III
STATISTICAL RESULTS OF $P(\mathbf{A})$ (%)

Profile	Sequence	QP=22	QP=27	QP=32	QP=37	Average
RAM	BQSquare	61.06	62.51	63.82	65.28	63.17
	BasketballDrill	67.97	70.79	73.46	75.43	71.91
	Johnny	n/a	n/a	n/a	n/a	n/a
	Cactus	63.73	69.46	72.85	75.26	70.33
	Traffic	69.73	71.10	72.62	73.23	71.67
	Average	65.62	68.47	70.69	72.30	69.27
LDM	BQSquare	55.90	57.52	59.09	60.23	58.19
	BasketballDrill	61.46	63.72	66.84	70.62	65.66
	Johnny	75.50	79.23	81.97	85.30	80.50
	Cactus	59.96	66.21	70.15	74.52	67.71
	Traffic	n/a	n/a	n/a	n/a	n/a
	Average	63.21	66.67	69.51	72.67	68.01

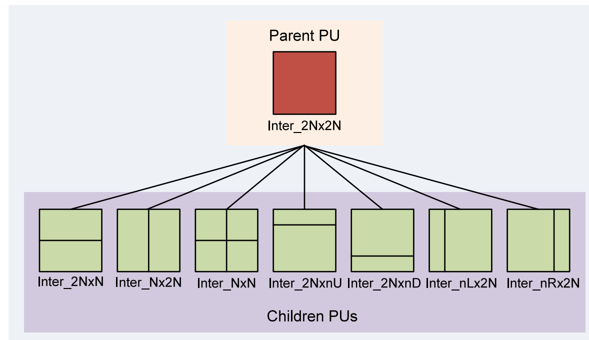


Fig. 3. An illustration on the parent PU and its children PUs.

global search point. In addition, we can see that the probability $P(\mathbf{A})$ of the video sequence "Johnny" is more than 80.50% on average for the LDM coding profile. This is because that the "Johnny" has most of the regions with static background, and these regions have strong spatial correlation. From these values, we can draw a conclusion that the proposed ISP decision method works efficiently for finding the final global optimal search point with low complexity in the ME process.

2) Content Similarity Based Block Matching Search Skipped Strategy: In the H.265/HEVC PU encoding process, the CU is further partitioned into PUs according to the PU prediction-type. For example, a CU can be split into a PU with size of 2N×2N, two PUs with size of 2N×N, N×2N, 2N×nU, 2N×nD, nL×2N, or nR×2N, or four PUs with size of N×N. In the ME process, all inter PUs (Inter_2N×2N, Inter_2N×N, Inter_N×2N, Inter_N×N, Inter_2N×nU, Inter_2N×nD, Inter_nL×2N, and Inter_nR×2N) need to perform block matching search in the reference frame to finding the optimal matching block. Thus, based on the PU size, the Inter_2N×2N PU mode can be regarded as the parent of the other inter PU modes due to the content of children PUs is a part of their parent PU, an example of the parent PU and children PUs is shown in Fig. 3. In the ME process, if the parent PU selects the ISP as its final optimal search point, it represents that the content of the parent PU with size of 2N×2N is a static region, with simple texture, or moves uniform. In this circumstance, the children PUs may also have a large probability to select the ISP as their best search point due to the high spatial correlation and similar properties of the pixels within a CU.

TABLE IV
STATISTICAL RESULTS OF THE CONDITIONAL PROBABILITY $P(\mathbf{C}|\mathbf{B})$ (%)

Profile	Sequence	QP=22	QP=27	QP=32	QP=37	Average
RAM	BQSquare	93.97	94.22	94.47	94.84	94.38
	BasketballDrill	91.90	92.16	92.95	93.92	92.73
	Johnny	n/a	n/a	n/a	n/a	n/a
	Cactus	89.67	91.34	92.69	93.88	91.90
	Traffic	93.59	94.08	94.60	94.84	94.28
	Average	92.28	92.95	93.68	94.37	93.32
LDM	BQSquare	97.16	97.12	97.01	97.04	97.08
	BasketballDrill	91.33	91.71	92.77	94.26	92.52
	Johnny	94.42	95.45	96.44	97.27	95.90
	Cactus	88.32	90.19	91.53	93.27	90.83
	Traffic	n/a	n/a	n/a	n/a	n/a
	Average	92.81	93.62	94.44	95.46	94.08

In order to analyze the final optimal search point decision correlation among the parent PU and its children PUs, let the event **B** denotes that the parent PU selects the ISP as its final best point, the event **C** represents that the children PUs choose the ISP as their final best point. Five video sequences are tested under the conditions listed in Table I. The statistical results of conditional probability $P(\mathbf{C}|\mathbf{B})$ are tabulated in Table IV.

From Table IV, it can be seen that for the RAM coding profile, the probability of $P(\mathbf{C}|\mathbf{B})$ is from 89.67% to 94.84%, 93.32% on average. For the LDM coding profile, the probability of $P(\mathbf{C}|\mathbf{B})$ is from 88.32% to 97.27%, 94.08% on average. In addition, we can see that for the video sequence "Cactus", the probability of $P(\mathbf{C}|\mathbf{B})$ has a little decrease, 91.90% and 90.83% on average, for the RAM and LDM coding profiles, respectively. This is because that the content of "Cactus" has complexity texture, which results in the spatial MV correlation decreases. Moreover, the probability of $P(\mathbf{C}|\mathbf{B})$ becomes large as the QP increases, the reason is that when the QP increases, the distortions will be transformed and quantized into small values, which results in the large spatial MV correlation between the parent PU and its children PUs. From these values, we can figure out that if the parent PU selects the ISP as its best point, its children PUs also have a rather large probability to choose the ISP as their final optimal search point. As a results, the block matching search of the children PUs could be skipped, and significant encoding time saving could be gained.

Based on above analyses, the final optimal search point of the children PU, $\vec{MV}_{children}$, is determined by Eq. (6), as shown at the bottom of this page, where $\vec{MV}_{CISP}$ is the MV of the ISP of the children PUs; $\vec{MV}_{parent}$ represents the MV of the final optimal search point of the parent PU; $\vec{MV}_{PISP}$ indicates the MV of the ISP of the parent PU; Ω is the set of the candidate MVs of the block matching search points of the children PUs.

3) The Overall Method: Based on the above analyses, the proposed fast ME method is summarized step-by-step as follows, which is also illustrated in Fig. 4.

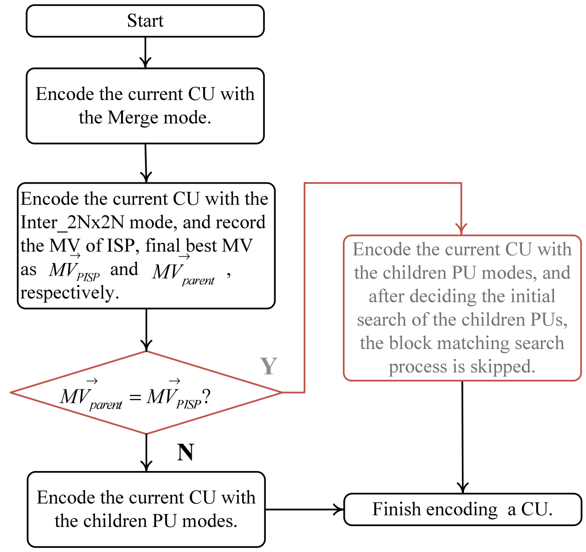


Fig. 4. The flowchart of the proposed fast ME method.

Step 1. Encode the current CU with the Merge/SKip mode, the parent PU mode, Inter_2N×2N mode, then record the ISP and final best search point of the parent PU mode as $\vec{MV}_{PISP}$, and $\vec{MV}_{parent}$, respectively. If $\vec{MV}_{parent} = \vec{MV}_{PISP}$, go to Step 2. Otherwise, go to Step 3.

Step 2. Encode the current CU with the children PU modes, including Inter_2N×N, Inter_N×2N, Inter_N×N, Inter_2N×nU, Inter_2N×nD, Inter_nL×2N, and Inter_nR×2N. After deciding the initial search point of these children PU modes, the block matching search process is skipped for these modes, and the initial search point is set as the final optimal search point. Go to Step 4.

Step 3. Encode the current CU with the inter PU modes, including Inter_2N×N, Inter_N×2N, Inter_N×N, Inter_2N×nU, Inter_2N×nD, Inter_nL×2N, and Inter_nR×2N. Go to Step 4.

Step 4. The Intra PU modes are used for encoding the current CU. Go to Step 5;

Step 5. Store the coding information and write the encoded bit stream. Go back to Step 1 to process the next CU.

IV. EXPERIMENTAL RESULTS

A. Encoding Complexity Saving Analyses on the Proposed Method

From the overall method, we can see that the computational complexity saving of the proposed fast ME method depends on the number of the parent PUs which select the ISP as their final optimal search point. If most of the PUs select the ISP as the final best point, then the computational complexity saving of the proposed fast ME method will be large. Otherwise, the computational complexity saving of the ME will be small.

$$\vec{MV}_{children} = \begin{cases} \vec{MV}_{CISP}, & if \;\; \vec{MV}_{parent} = \vec{MV}_{PISP}, \\ \arg\min_{\vec{MV}_n} J\left(\vec{MV}_n, \lambda_{MOTION}\right), s.t. \vec{MV}_n \in \left(\Omega \cup \vec{MV}_{CISP}\right), & otherwise, \end{cases} \tag{6}$$

TABLE V
PERCENTAGE OF THE ISP TO BE SELECTED AS THE FINAL OPTIMAL SEARCH POINT OF THE PARENT PU (%)

Profile	Sequence	QP=22	QP=27	QP=32	QP=37	Average
RAM	BQSquare	88.97	89.81	90.51	91.16	90.11
	BasketballDrill	84.89	86.86	88.95	90.56	87.82
	Johnny	n/a	n/a	n/a	n/a	n/a
	Cactus	82.42	86.11	88.76	90.64	86.98
	Traffic	89.09	89.81	90.25	90.64	89.95
	Average	86.34	88.15	89.62	90.75	88.72
LDM	BQSquare	96.09	96.20	96.26	96.17	96.18
	BasketballDrill	85.63	87.56	89.61	91.70	88.63
	Johnny	92.01	93.48	94.66	96.06	94.05
	Cactus	80.58	84.39	87.03	89.06	85.27
	Traffic	n/a	n/a	n/a	n/a	n/a
	Average	88.58	90.41	91.89	93.25	91.03

To analyze the theoretical values of how many children PUs skip the block matching search, five video sequences are tested in real video coding. The test conditions are listed in Table I, and the percentage of the parent PUs which choose the ISP as their final optimal search point is tabulated in Table V.

From Table V, it can be seen that most of the parent PUs select the ISP as their final optimal search point in the ME process. For the RAM coding profile, there are from 82.42% to 91.16%, 88.72% on average, parent PUs selecting the ISP as their final best search point. For the LDM coding profile, there are about 80.58% to 96.26%, 91.03% on average, parent PUs choosing the ISP as their final optimal search point. In other words, there are about 88.72% and 91.03% children PUs skipped the block matching search process in the ME process for the RAM and LDM coding profiles, respectively. From these values, we can draw a conclusion that the proposed fast ME method can efficiently reduce the computational complexity of the ME process of the H.265/HEVC encoder.

B. Encoding Performance on PSNR, BR, ME Time Saving, and Total Encoding Time Saving

In order to prove the efficiency of the proposed fast ME method, the testing environment is defined as follows: the software platform is the HEVC reference software HM12.0; the hardware platform is Intel Xeon CPU E5-1620 v2 @ 3.70GHz, 16.0GB RAM with the Microsoft Windows 7 64-bit operating system. The test conditions are presented in Table I. The other coding parameters adopt the default settings in the HM12.0 and coding profiles.

We compared the proposed fast ME algorithm with two recently published methods, ETZSearch [20] and ISOCC [28], in terms of peak signal-to-noise ratio (PSNR), bit rate (BR), total encoding time saving (TETS), and ME time saving (METS). The benchmark is the HM12.0, and the comparison results are tabulated in Table VI, where the Bjontegaard delta PSNR (BDPSNR) and Bjontegaard delta BR (BDBR) are the average PSNR differences in dB for the same BR, and the average BR differences in percent for the same PSNR, respectively, and are computed according [29]; TETS and METS are calculated as

$$\begin{cases} TETS = \dfrac{1}{4}\displaystyle\sum_{i=1}^{4} \dfrac{T_{\psi}(QP_i) - T_{HM}(QP_i)}{T_{HM}(QP_i)} \times 100\%, \\ METS = \dfrac{1}{4}\displaystyle\sum_{i=1}^{4} \dfrac{TE_{\psi}(QP_i) - TE_{HM}(QP_i)}{TE_{HM}(QP_i)} \times 100\%, \end{cases} \tag{7}$$

where $T_{\psi}(QP_i)$ denotes the total encoding time of the HM12.0 with the fast ME method ψ under the QP value of QP_i, $\psi \in$ {ETZSearch [20], ISOCC [28], Proposed}; $T_{HM}(QP_i)$ is the total encoding time of the HM12.0 with the original ME method, TZSearch, under the QP value of QP_i; $TE_{\psi}(QP_i)$ indicates the total ME time of HM12.0 with the fast ME method ψ under the QP value of QP_i, $\psi \in$ {ETZSearch [20], ISOCC [28], Proposed}; $TE_{HM}(QP_i)$ is the total ME time of the HM12.0 with the original fast ME method, TZSearch, under the QP value of QP_i; QP_i={22, 27, 32, 37}. The original fast ME algorithm, TZSearch which is used in the HM12.0, is denoted as HM-TZSearch.

From Table VI, it can be observed that for the LDM coding profile, the ETZSearch reduces the total encoding time from 1.18% to 5.48%, 2.95% on average; and saves the ME time from 1.40% to 8.15%, 4.30% on average. Meanwhile, the BDPSNR between the ETZSearch and the original HM-TZSearch is from -0.009 dB to -0.001 dB, -0.004 dB on average; and the BDBR between the ETZSearch and the original HM-TZSearch is from 0.01% to 0.29%, 0.12% on average. For the RAM coding profile, the ETZSearch saves the total encoding time from 1.43% to 4.86%, 2.79% on average; and reduces the ME time from 2.19% to 7.02%, 4.25% on average. While the BDPSNR between the ETZSearch and the original HM-TZSearch is from -0.014 dB to -0.001 dB, -0.005 dB on average; the BDBR between the ETZSearch and the original HM-TZSearch is from 0.06% to 0.39%, 0.14% on average. From these value, we can see that the ETZSearch can efficiently maintain the RD performance of the original HM12.0, whereas the total encoding time saving and ME time saving of the ETZSearch are quite limited. The reasons of low time saving are that 1) the ETZSearch uses a four points diamond search to early terminate the ME process, it also consumes extra encoding time for checking the four points; 2) the original HM-TZSearch adopts some early termination strategies for the block matching search process, which results in the ETZSearch doesn't work in the HM-TZSearch.

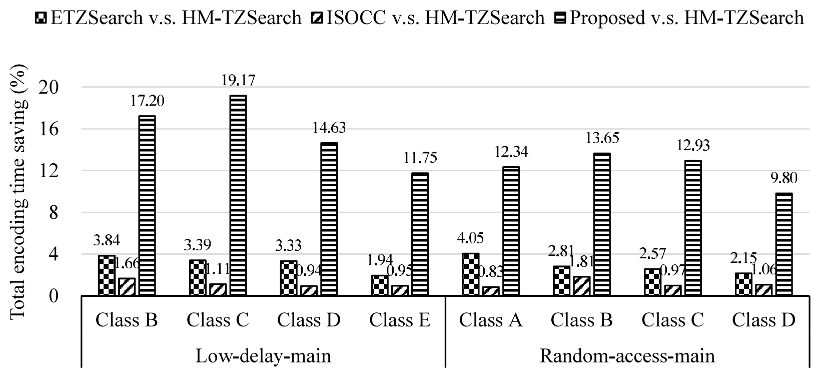
For the LDM coding profile, the ISOCC only achieves 1.15% and 1.53% of the average total encoding time saving and average ME time saving, respectively. The average BDPSNR and BDBR between the ISOCC and the original HM-TZSearch are -0.006 dB and 0.18%, respectively. For the RAM coding profile, the ISOCC reduces the total encoding time and ME encoding time for 1.25% and 1.68% on average, respectively. The average BDPSNR and BDBR between the ISOCC and the original HM-TZSearch are -0.006 dB and 0.17%, respectively. From these values, we can seen that the complexity saving of the ISOCC is quite limited. This is because that the ISOCC only refines the block matching search pattern of the HM-TZSearch, while most of the block matching search processes are skipped in the HM-TZSearch encoding process.

TABLE VI
SUMMARY OF ENCODING RESULTS

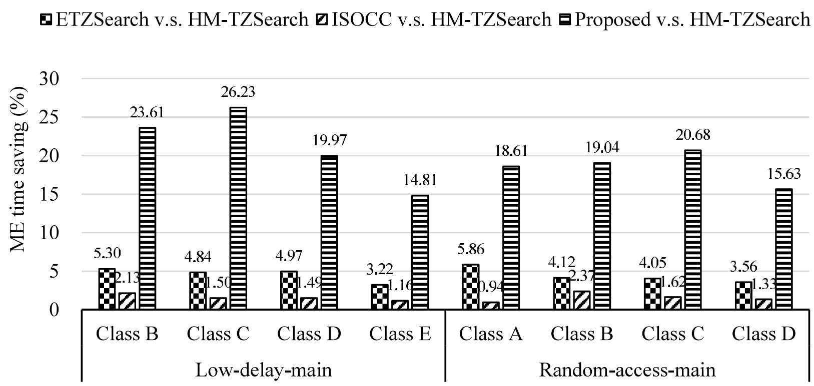
Algorithm	Class	Resolution	Sequence	Low-delay-main coding profile				Random-access-main coding profile			
				BDPSNR (dB)	BDBR (%)	TETS (%)	METS (%)	BDPSNR (dB)	BDBR (%)	TETS (%)	METS (%)
ETZSearch [20] v.s. HM-TZSearch	A	2560×1600	Traffic	n/a[2]	n/a	n/a	n/a	-0.014	0.39	-3.98	-5.53
			PeopleOnStreet	n/a	n/a	n/a	n/a	-0.009	0.20	-4.11	-6.19
	B	1920×1080	ParkScene	-0.003	0.09	-2.36	-3.35	-0.002	0.07	-1.84	-3.03
			Cactus	-0.004	0.13	-4.31	-5.45	-0.001	0.06	-2.96	-4.24
			BQTerrace	-0.002	0.09	-2.24	-3.08	-0.002	0.10	-1.57	-2.19
			BasketballDrive	-0.001	0.01	-6.44	-8.15	-0.002	0.10	-4.86	-7.02
	C	832×480	BQMall	-0.002	0.04	-4.52	-6.49	-0.008	0.20	-3.14	-4.79
			PartyScene	-0.001	0.03	-2.36	-3.35	-0.005	0.10	-1.84	-3.03
			BasketballDrill	-0.007	0.19	-3.30	-4.68	-0.001	0.03	-2.72	-4.34
	D	416×240	BQSquare	-0.001	0.03	-2.18	-4.23	-0.003	0.07	-1.87	-3.06
			BlowingBubbles	-0.002	0.07	-2.32	-3.81	-0.007	0.17	-1.43	-2.45
			BasketballPass	-0.002	0.04	-5.48	-6.87	-0.008	0.14	-3.14	-5.17
	E	1280×720	FourPeople	-0.002	0.05	-1.65	-3.92	n/a	n/a	n/a	n/a
			Johnny	-0.008	0.15	-1.99	-4.39	n/a	n/a	n/a	n/a
			KristenAndSara	-0.006	0.27	-2.68	-4.10	n/a	n/a	n/a	n/a
			Vidyo1	-0.007	0.21	-1.18	-1.40	n/a	n/a	n/a	n/a
			Vidyo3	-0.009	0.29	-1.36	-1.84	n/a	n/a	n/a	n/a
			Vidyo4	-0.007	0.17	-2.77	-3.66	n/a	n/a	n/a	n/a
	Average			**-0.004**	**0.12**	**-2.95**	**-4.30**	**-0.005**	**0.14**	**-2.79**	**-4.25**
ISOCC [28] v.s. HM-TZSearch	A	2560×1600	Traffic	n/a	n/a	n/a	n/a	-0.002	0.05	-0.89	-1.19
			PeopleOnStreet	n/a	n/a	n/a	n/a	-0.017	0.39	-0.77	-0.69
	B	1920×1080	ParkScene	-0.007	0.22	-0.47	-0.82	-0.004	0.12	-1.14	-1.20
			Cactus	-0.002	0.08	-0.92	-1.20	-0.005	0.19	-1.14	-1.20
			BQTerrace	-0.003	0.14	-3.48	-4.25	-0.006	0.31	-3.52	-5.15
			BasketballDrive	-0.004	0.16	-1.75	-2.24	-0.002	0.17	-1.45	-1.93
	C	832×480	BQMall	-0.008	0.19	-1.38	-1.41	-0.001	0.03	-1.20	-1.83
			PartyScene	-0.003	0.08	-0.98	-1.45	-0.009	0.20	-0.82	-1.47
			BasketballDrill	-0.002	0.04	-0.98	-1.63	-0.004	0.11	-0.90	-1.56
	D	416×240	BQSquare	-0.005	0.11	-0.81	-1.61	-0.010	0.21	-1.06	-1.75
			BlowingBubbles	-0.004	0.11	-1.14	-1.87	-0.006	0.14	-1.07	-1.33
			BasketballPass	-0.029	0.62	-0.87	-1.00	-0.008	0.16	-1.06	-0.90
	E	1280×720	FourPeople	-0.005	0.03	-0.73	-1.01	n/a	n/a	n/a	n/a
			Johnny	-0.005	0.22	-1.09	-1.17	n/a	n/a	n/a	n/a
			KristenAndSara	-0.006	0.23	-1.06	-1.17	n/a	n/a	n/a	n/a
			Vidyo1	-0.003	0.16	-1.01	-1.17	n/a	n/a	n/a	n/a
			Vidyo3	-0.007	0.18	-0.92	-1.28	n/a	n/a	n/a	n/a
			Vidyo4	-0.005	0.30	-0.87	-1.14	n/a	n/a	n/a	n/a
	Average			**-0.006**	**0.18**	**-1.15**	**-1.53**	**-0.006**	**0.17**	**-1.25**	**-1.68**
Proposed v.s. HM-TZSearch	A	2560×1600	Traffic	n/a	n/a	n/a	n/a	-0.015	0.41	-8.23	-11.52
			PeopleOnStreet	n/a	n/a	n/a	n/a	-0.071	1.58	-16.44	-25.69
	B	1920×1080	ParkScene	-0.020	0.63	-13.54	-20.45	-0.022	0.68	-9.64	-14.23
			Cactus	-0.012	0.49	-17.82	-22.35	-0.017	0.73	-12.90	-19.45
			BQTerrace	-0.006	0.29	-11.29	-17.95	-0.008	0.38	-10.99	-14.85
			BasketballDrive	-0.010	0.43	-26.14	-33.68	-0.015	0.72	-21.08	-27.62
	C	832×480	BQMall	-0.038	0.92	-23.36	-28.81	-0.043	1.05	-13.07	-20.74
			PartyScene	-0.030	0.67	-14.16	-22.35	-0.039	0.84	-11.01	-18.36
			BasketballDrill	-0.038	0.96	-19.98	-27.53	-0.031	0.76	-14.71	-22.94
	D	416×240	BQSquare	-0.017	0.40	-8.54	-13.35	-0.036	0.80	-7.28	-11.06
			BlowingBubbles	-0.023	0.6	-13.99	-17.65	-0.037	0.91	-8.90	-14.65
			BasketballPass	-0.053	1.12	-21.36	-28.91	-0.070	1.46	-13.21	-21.17
	E	1280×720	FourPeople	-0.009	0.20	-10.12	-13.69	n/a	n/a	n/a	n/a
			Johnny	-0.008	0.21	-10.57	-14.12	n/a	n/a	n/a	n/a
			KristenAndSara	-0.018	0.50	-11.77	-14.71	n/a	n/a	n/a	n/a
			Vidyo1	-0.008	0.28	-11.21	-13.56	n/a	n/a	n/a	n/a
			Vidyo3	-0.017	0.57	-12.48	-14.53	n/a	n/a	n/a	n/a
			Vidyo4	-0.011	0.54	-14.32	-18.25	n/a	n/a	n/a	n/a
	Average			**-0.020**	**0.55**	**-15.04**	**-20.12**	**-0.034**	**0.86**	**-12.29**	**-18.52**

It can also be seen that the proposed method efficiently removes the encoding complexity of the ME of the HEVC encoder, while maintaining a comparable RD performance. For the LDM coding profile, the proposed method reduces the total encoding time from 10.12% to 26.14%, 15.04% on average; and saves the ME time from 13.35% to 33.68%, 20.12% on average. Meanwhile, the BDPSNR between the proposed method and the original HM-TZSearch is from -0.053 dB to -0.006 dB, -0.020 dB on average; and the BDBR between the proposed method and the original HM-TZSearch is from 0.21% to 1.12%, 0.55% on average. For the RAM coding profile, the proposed method reduces the encoding time from 7.28% to 21.08%, 12.29% on average; and achieves the ME time saving from 11.06% to 27.62%, 18.52% on average. At the same time, the BDPSNR between the proposed method and the original HM-TZSearch is from -0.070 dB to -0.008 dB, -0.034 dB on average; and the BDBR between the proposed

[2]The "n/a" indicates the result is not available, since the sequences of class A with RAM coding profile, and the sequences of class E with LDM coding profiles are not tested according to the HEVC common test conditions [26], [27].



(a)



(b)

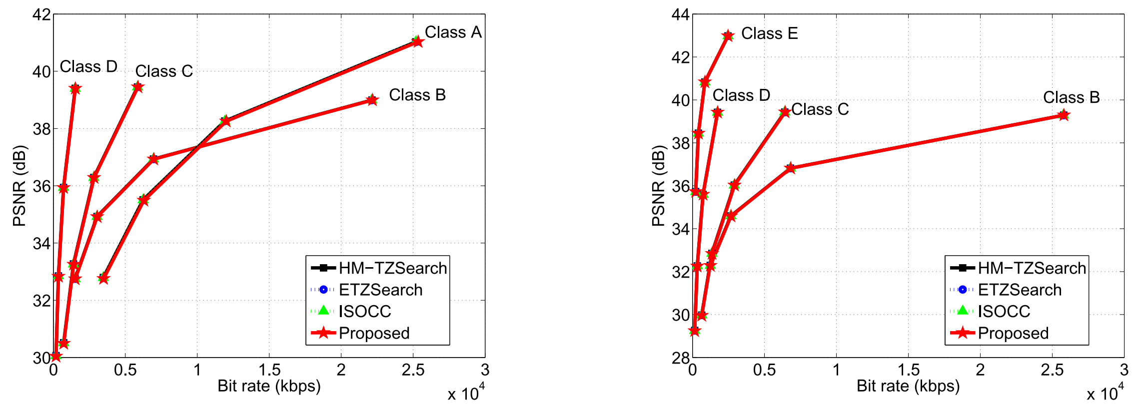
Fig. 5. Summary of encoding time saving. (a) Total encoding time saving of the HEVC encoder. (b) Total encoding time saving of the HEVC ME process.

method and the original HM-TZSearch is from 0.38% to 1.58%, 0.86% on average.

From these values, we can observe that the TETS and TMES of the proposed algorithm with LDM coding profile are larger than that of the RAM coding profile, this is because that the encoding complexity of the LDM coding profile is more complex than that of the RAM coding profile, which results in more encoding complexity could be removed in the LDM coding profile. In addition, the proposed method removes more encoding complexity for the video sequences, “BasketballDrill”, “BasketballPass”, and “PeopleOnStreet”, this is because that these videos have violent motion activity content, which results in the block matching search consumes much more time to locate the final best search point. Thus, the block matching search process is skipped for these video sequences, significant encoding time saving is obtained.

In order to show the time saving efficiency of the proposed method, the encoding time saving comparison among the ETZSearch, the ISOCC, and the proposed method is presented in Fig. 5. It can be seen that the proposed method can efficiently reduce the encoding time of the H.265/HEVC encoder and the encoding time of the ME process. For the LDM coding profile, the average total encoding time saving of the ETZSearch, ISOCC, and the proposed method are 2.95%, 1.15% and 15.04%, respectively; and the average ME time saving of the ETZSearch, ISOCC, and the proposed method are 4.30%, 1.53%, and 20.12%, respectively. For the RAM coding profile, the average total encoding time saving of the ETZSearch, ISOCC, and the proposed method are 2.79%, 1.25%, and 12.29%, respectively; and the average ME time saving of the ETZSearch, ISOCC, and the proposed method are 4.25%, 1.68%, and 28.52%, respectively. Compared to the ETZSearch, the proposed method saves about 12.46% and 9.77% total encoding time for the LDM and RAM coding profiles, respectively; and the proposed method reduces about 16.53% and 14.90% encoding time for the ME process of the H.265/HEVC encoder with LDM and RAM coding profiles, respectively. For the LDM coding profile, the proposed method can reduce 14.05% total encoding time and 18.88% ME encoding time when compared to the ISOCC; the proposed method saves 11.18% total encoding time and 16.85% ME encoding time for the RAM coding profile when compared to the ISOCC. From these values, we can draw a conclusion that the proposed method reduces the encoding complexity of the ME process of the H.265/HEVC encoder more efficient than the ETZSearch and ISOCC.

To intuitively explain the RD performance, the RD curves of classes A, B, C, and D, with RAM coding profiles, and the RD curves of classes B, C, D, and E are presented in Fig. 6 (a) and Fig. 6 (b), respectively. It can be seen from Fig. 6 that the proposed fast ME method obtains almost the same RD performance with the HM-TZSearch, ETZSearch, and ISOCC. From the above data, we conclude that the proposed method efficiently removes the encoding complexity of the integer-pixel ME of the H.265/HEVC, meanwhile, the RD performance degradation is negligible.



(a)

(b)

Fig. 6. RD curves. (a) RAM coding profile. (b) LDM coding profile.

TABLE VII
THE FALSE DECISION RATES OF THE BEST MV (%)

Profile	Sequence	ETZSearch	ISOCC	Proposed
RAM	BQSquare	4.97	5.08	5.63
	BasketballDrill	5.46	6.39	7.27
	Johnny	n/a	n/a	n/a
	Cactus	7.05	7.62	8.11
	Traffic	5.41	4.16	5.72
	Average	5.72	5.81	6.68
LDM	BQSquare	2.01	2.35	2.92
	BasketballDrill	6.68	6.23	7.48
	Johnny	3.54	4.03	4.11
	Cactus	8.47	7.68	9.17
	Traffic	n/a	n/a	n/a
	Average	5.18	5.07	5.92

C. The False Decision Rates of the Best Motion Vector

In order to evaluate the best MV decision efficiency of the proposed fast ME method, the false decision rates of the best MV of the ETZSearch, ISOCC, and our proposed method are evaluated with five HEVC standard sequences, as "BQSquare", "BasketballDrill", "Johny", "Cactus", and "Traffic", under the RAM and LDM coding profiles, respectively. Table VII presents the test results of the average best MV false decision rate.

From Table VII, it can be seen that for the RAM coding profile, the average false decision rates of the best MV of the ETZSearch, ISOCC, and our proposed method is 5.72%, 5.81%, and 6.68%, respectively. For the LDM coding profile, the average false decision rates of the best MV of the ETZSearch, ISOCC, and our proposed method is 5.18%, 5.07%, and 5.92%, respectively. From these results, we can see that the false decision rates of the best MV of our proposed method is very close to the ETZSearch and ISOCC, while its value is within an acceptable range.

V. CONCLUSION

To make the H.265/HEVC meet with the requirements of the current broadcasting applications, e.g., low complexity, high compression, we proposed a fast ME method to reduce the computational complexity of the H.265/HEVC in this paper. Firstly, based on the prediction type, all PUs are classified into two categories, i.e., parent PU and children PUs. Then, based on the best MV selection correlation between the parent PU and its children PUs, the block matching search process of the children PUs in ME process is adaptively skipped if their parent PU selects the ISP as its final best MV. Experimental results show that the proposed fast ME method efficiently removes the computational complexity of the ME process of the H.265/HEVC encoder, while maintaining comparable RD performance.

REFERENCES

[1] B. Chen et al., "Color image analysis by quaternion-type moments," J. Math. Imag. Vis., vol. 51, no. 1, pp. 124–144, Jan. 2015.
[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[3] "Information technology-high efficiency coding and media delivery in heterogeneous environments—Part 2: High efficiency video coding," ISO/IEC 23008-2:2013 ITU-T Rec. H.265, 2013.
[4] J. Lei, K. Feng, M. Wu, S. Li, and C. Hou, "Rate control of hierarchical B prediction structure for multi-view video coding," Multimedia Tools Appl., vol. 72, no. 1, pp. 825–842, Sep. 2014.
[5] Z. Pan, S. Kwong, Y. Zhang, J. Lei, and H. Yuan, "Fast coding tree unit depth decision for high efficiency video coding," in Proc. IEEE Int. Conf. Image Process. (ICIP), Paris, France, Oct. 2014, pp. 3214–3218.
[6] 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.
[7] Y. Zhang et al., "Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding," IEEE Trans. Image Process., vol. 24, no. 7, pp. 2225–2238, Jul. 2015.
[8] 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.
[9] 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.
[10] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding," IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 3, pp. 422–435, Mar. 2015.
[11] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Lshiguro, "Motioncompensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., New Orleans, LA, USA, 1981, pp. G5.3.1–G5.3.5.
[12] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 313–317, Jun. 1996.

[13] 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.

[14] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 5, pp. 349–355, May 2002.

[15] F. Bossen, D. Flynn, and K. Suehring, *JCT-VC AHG Report: HEVC HM Software Development and Software Technical Evaluation (AHG3)*, document JCTVC-O0003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Oct. 2013.

[16] Z. Pan, S. Kwong, L. Xu, Y. Zhang, and T. Zhao, "Predictive and distribution-oriented fast motion estimation for H.264/AVC," *J. Real Time Image Process.*, vol. 9, no. 4, pp. 597–607, Dec. 2014.

[17] Z. Pan, Y. Zhang, and S. Kwong, "Efficient motion and disparity estimation optimization for low complexity multiview video coding," *IEEE Trans. Broadcast.*, vol. 61, no. 2, pp. 166–176, Jun. 2015.

[18] N. Purnachand, L. N. Alves, and A. Navarro, "Fast motion estimation algorithm for HEVC," in *Proc. Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*, Berlin, Germany, Sep. 2012, pp. 34–37.

[19] X. Li, R. Wang, W. Wang, Z. Wang, and S. Dong, "Fast motion estimation methods for HEVC," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Beijing, China, Jun. 2014, pp. 1–4.

[20] Z. Pan, Y. Zhang, S. Kwong, X. Wang, and L. Xu, "Early termination for TZSearch in HEVC motion estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 1389–1393.

[21] X. Li, R. Wang, X. Cui, and W. Wang, "Context-adaptive fast motion estimation of HEVC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 2784–2787.

[22] S.-H. Yang, J.-Z. Jiang, and H.-J. Yang, "Fast motion estimation for HEVC with directional search," *Electron. Lett.*, vol. 50. no. 9, pp. 673–675, Apr. 2014.

[23] Z.-T. Liao and C.-A. Shen, "A novel search window selection scheme for the motion estimation of HEVC systems," in *Proc. Int. SoC Design Conf. (ISOCC)*, Gyeongju, South Korea, Nov. 2015, pp. 267–268.

[24] Y. Li, Y. Liu, H. Yang, and D. Yang, "An adaptive search range method for HEVC with the k-nearest neighbor algorithm," in *Proc. Vis. Commun. Image Process. (VCIP)*, Singapore, Dec. 2015, pp. 1–4.

[25] J.-R. 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.

[26] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-J1100, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Mar. 2012.

[27] F. Bossen, *Common Test Conditions and Software Reference Configurations*, document JCTVC-D600, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, Jan. 2011.

[28] N. Parmar and M. H. Sunwoo, "Enhanced test zone search motion estimation algorithm for HEVC," in *Proc. Int. SoC Design Conf. (ISOCC)*, Jeju, South Korea, Nov. 2014, pp. 260–261.

[29] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33, ITU-T VCEG, Geneva, Switzerland, Apr. 2001.

Zhaoqing Pan (M'14) received the B.S. degree in computer science and technology from Yancheng Normal University, Yancheng, China, in 2009, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2014. In 2013, he was a Visiting Scholar with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, for six months. He is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interest focuses on low-complexity coding algorithms for video compression.

Jianjun Lei (M'11) received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007. He was a Visiting Researcher with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, from 2012 to 2013. He is currently a Professor with the School of Electronic Information Engineering, Tianjin University, Tianjin, China. His research interests include 3-D video processing, 3-D display, and computer vision.

Yun Zhang (M'12–SM'16) received the B.S. and M.S. degrees in electrical engineering from Ningbo University, Ningbo, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2009 to 2014, he was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong, Hong Kong. In 2010, he became an Assistant Professor with the Shenzhen Institutes of Advanced Technology, CAS, Shenzhen, China, where he has been an Associate Professor, since 2012. His research interests are video compression, 3-D video processing, and visual perception.

Xingming Sun (SM'07) received the B.S. degree in mathematics from Hunan Normal University, Changsha, China, in 1984, the M.S. degree in computing science from the Dalian University of Science and Technology, Dalian, China, in 1988, and the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2001. He is currently a Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include image processing and information security.

Sam Kwong (M'93–SM'04–F'13) received the B.S. degree from the State University of New York at Buffalo in 1983, the M.S. degree from the University of Waterloo, Waterloo, ON, Canada, in 1985, both in electrical engineering, and the Ph.D. degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data Canada. He joined Bell Northern Research Canada as a Member of Scientific Staff. In 1990, he became a Lecturer with the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor with the Department of Computer Science. His research interests are video and image coding and evolutionary algorithms.