

# Fast encoding algorithm for high-efficiency video coding (HEVC) system based on spatio-temporal correlation

Jong-Hyeok Lee · Kalyan Goswami ·  
Byung-Gyu Kim · Seyoon Jeong · Jin Soo Choi

Received: 25 September 2014 / Accepted: 30 December 2014 / Published online: 13 January 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** Video compression technology is an important research part to the intelligent user interface for interactive multimedia system using technologies and services such as image processing, pattern recognition, computer vision and cloud computing service. Recently, high-efficiency video coding (HEVC) has been established as the demand of very high-quality multimedia service like ultrahigh definition video service. High-efficiency video coding (HEVC) standard has three units such as coding unit (CU), prediction unit (PU) and transform unit. It has too many complexities to improve coding performance. We propose a fast algorithm which can be possible to apply for both CU and PU parts. To reduce the computational complexity, we propose CU splitting algorithm based on rate–distortion cost of CU about the parent and current levels to terminate the CU decision early. In terms of PU, we develop fast PU decision based on spatio-temporal and depth correlation for PU level. Finally, experimental results show that our algorithm provides a significant time reduction for encoding with a small loss in video quality, compared to the

original HEVC Test Model (HM) version 10.0 software and the previous algorithm.

**Keywords** HEVC · Coding tree block (CTB) · Coding unit (CU) · Prediction unit (PU) · CU splitting · PU decision

## 1 Introduction

An intelligent user interface for interactive multimedia is an emerging field of study with the rapid development of hardware devices such as multidimensional sensors and motion detector [1]. The interactive multimedia means the integration of digital media including combinations of electronic text, graphics, moving images, and sound into a structured digital computerized environment that allows people to interact with the data in internet, telecoms and interactive digital television. There are many basic techniques and services such as image processing, pattern recognition, computer vision, storage/transmission for multimedia toward intelligent user interface, semantic web and smart grid services. This paper focuses on video coding technique to help the improvement of the performance in various multimedia services for interactive 3D video processing and cloud computing [2, 3, 4].

A new generation video coding standard, called a high-efficiency video coding (HEVC) [5], has been developed by the Joint Collaborative Team on Video Coding (JCT-VC) group. The JCT-VC is a group of video coding experts created by ITU-T Study Group 16 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG) in 2010.

Although coding efficiency improvement is solved it has more computational complexity than the H.264/AVC [6] because of tools having high complexity and high

---

J.-H. Lee · K. Goswami · B.-G. Kim (✉)  
Department of Computer Engineering, SunMoon University,  
Asan, Republic of Korea  
e-mail: bg.kim@ieee.org; bg.kim@mpcl.sunmoon.ac.kr

J.-H. Lee  
e-mail: ljh0607@mpcl.sunmoon.ac.kr

K. Goswami  
e-mail: kalyan\_goswami@mpcl.sunmoon.ac.kr

S. Jeong · J. S. Choi  
Realistic Media Research Section, ETRI, Daejeon,  
Republic of Korea  
e-mail: jsy@etri.re.kr

J. S. Choi  
e-mail: jschoi@etri.re.kr

resolution of sequences. Therefore, reduction of encoding time with convincing loss is an interesting issue.

The video encoding process in the HEVC has three units for block structure: (1) a coding unit (CU) is basic block unit like macroblock in the H.264/AVC, (2) a prediction unit (PU) for performing motion estimation, rate–distortion optimization and mode decision, (3) a transform unit (TU) for transform and entropy coding. Encoding process encodes from coding tree block (CTB) having largest block size to its 4 child CUs, recursively.

An early partition decision algorithm [7] has been presented that attempts to termination mode process after checking the inter mode with each of the PU partition modes. Kim et al. [8] proposed an early termination method based on average rate–distortion cost of the previous skipped CUs. Some fast algorithms have been reported that explore other components [9, 10]. To determine the CU size before the general RD optimization technique, the supported vector machine (SVM)-based CU splitting early termination algorithm is proposed [11]. Early determination algorithms have been proposed that use SKIP conditions from fast method decision schemes [12–14]. In [15], CU mode decision and TU size selection were reported using variance values of coding mode costs. A fast intraprediction approach based on gradient has been used in [16]. In [17], residual quadtree mode decision algorithm considering the rate–distortion efficiency has been reported.

A two-stage process approach is proposed in [18]. Wu et al. [19] reported method using spatial homogeneity and the temporal stationarity characteristics of video objects [19]. A direct prediction method based on interblock correlation and adaptive rate–distortion cost threshold to stop early was also proposed by Kim et al. [20]. In [21], a tree-pruning algorithm was proposed to make an early termination CU. Shen et al. [22] proposed an early CU size determination algorithm. Fast interprediction method to omit unnecessary process was proposed using probability of the best inter mode of  $nN \times nN$  PU [23]. If a inter  $2N \times 2N$  is skip mode as the best, advanced motion vector prediction (AMVP) mode is skipped. Ahn et al. [24] proposed a fast decision algorithm of CU partitioning based on sample adaptive offset (SAO) parameter values, PU and TU sizes, MV sizes, and coded block flag (cbf) information from the reference blocks and others to reduce the complexity of CU split decision.

In this paper, we propose an early CU and PU mode decision algorithm for reducing the computational complexity of mode decision process about CU and PU mode in the HEVC encoder. In a CTB, CU partitioning is performed. Firstly, based on the local statistic of the rate–distortion (RD) costs and motion vectors, an early CU splitting termination (CST) is proposed to decide whether a

CU should be decomposed into 4 lower dimension CUs or not. Then, block motion complexity that considered spatio-temporal PU information is defined to reduce early by fast PU decision.

This paper is organized as follows: In Chapter 2, a fast CU splitting termination (CST) and fast PU decision method are described for proposed algorithm. Chapter 3 presents the coding performance of the suggested algorithm combined by two methods. In Chapter 4, concluding remarks will be given.

## 2 Proposed work

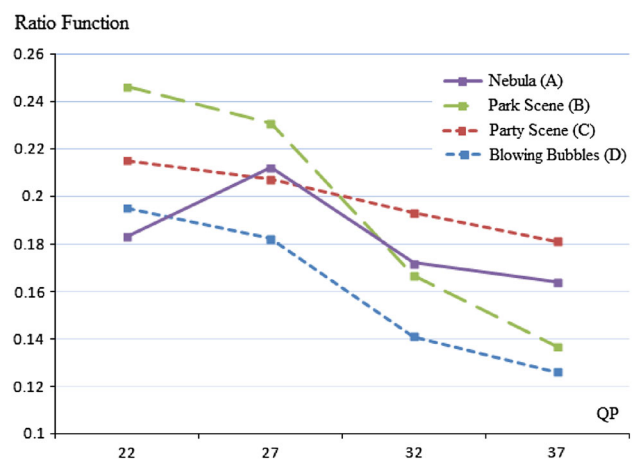
### 2.1 Fast coding unit splitting termination (CST) method

The main objective is to select the lowest RD cost before all combinations are finished. We have designed the ratio function of  $CU_d(i)$  at depth ‘ $d$ ’ that can be defined as:

$$r_i = \frac{RD_{cost}(CU_d(i))}{RD_{cost}(CU_{d-1})} \text{ where } \sum_{i=0}^3 r_i \leq 1. \quad (1)$$

When a CU is split, it will be divided up into 4 child CUs. Therefore, we can define a value of ratio function for each newly created CU. This parameter  $r_i$  denotes the defined ratio of the RD costs of the current CU and its parent CU. We have performed an analysis on the original HM reference software for 4 sequences with different dimensions. We have checked values of the ratio function for all encoded CUs for 4 QP values (i.e., 22, 27, 32, 37) which have been split in our experiment. As shown in Fig. 1, all the split CUs have smaller values of the ratio function than 0.25.

Therefore, when the ratio function of a child is lower than its siblings, it has low amount of chance to split in the



**Fig. 1** Ratio function analysis for 4 sequences between Class A to D

next depth level. Accordingly, this parameter can be used as a threshold for making the decision to split a current CU to next depth level.

After checking ratio function, we have incorporated different motion activities in a video sequence. In previous algorithm [21], if a CU is coded as SKIP mode then no further splitting of a CU. Motivating from this fact, we have explored other prediction modes of the PU. INTRA and INTER predictions have different kinds of PU types. INTER modes are 2 Nx2 N, 2 NxN, Nx2 N, NxN (presently, this mode is not used), 2 NxN, 2 NxN, nLx2 N and nRx2 N. INTRA modes are 2 Nx2 N and NxN. SKIP mode consisted of 2 Nx2 N size.

A CU is divided into 4 CUs of lower dimensions after the completion of all mode calculations. PU modes are processed recursively from the CTB to all children CUs. Table 1 has classified all the prediction modes according to motion activities. This kind of motion activity was explored by a good approach in the H.264/AVC codec [25, 27].

We have assumed that if a region in a video sequence have relatively slowly motion activity compared to the other regions, then there is a high chance that the slowly moving region will not split in that hierarchy of the CTB. From this concept, we have performed an experiment to show the percentage of non-splitting of a CU in any level of hierarchy of CTB. We have tested 4 sequences for different resolutions for 4 QP values in 20 frames and observed the CUs which are not split for different PU partition mode (PU\_WF).

Table 2 shows result of this experiment. From this table, it is clear that when PU\_WF is 0 (SKIP mode), there is a very high chance that the CU will not split in the next level. Hence, if the PU\_WF is 0, then we can directly decide that there is no need to split the CU. Also, if PU\_WF is 1 or 2, there is a quite good amount of chance that the CUs are still not split in the next level.

To investigate whether PU\_WF is 1 or 2, we have calculated a local average of the RD cost values of all encoded CUs. This parameter is not a static value as it takes the

values dynamically from the encoded CUs. In this process, the final PU mode and the corresponding RD cost values are checked after encoding CU. The Eqs. (2) and (3) are calculated after encoding of a CU check dimension (d) of CU and PU\_WF (wf) under satisfying  $wf < 3$ .

$$\text{sum}_d^{wf} = (\text{avg}_d^{wf} \times \text{count}_d^{wf}) + \text{RD cos } t_d^{wf}, \tag{2}$$

$$\text{avg}_d^{wf} = \frac{\text{sum}_d^{wf}}{\text{count}_d^{wf} + 1} \tag{3}$$

We have performed a test to calculate the CU non-splitting probability when the RD cost of the CU is less than its local average RD cost. Table 3 shows the tested result using 4 sequences having same environment. As shown in Table 3, it is clear that for PU\_WF 0, 1 and 2, the probabilities are almost within 65–75 %. It means that the proposed model can be effective when encoding the current CU. Also, an absolute motion vector (MV) is also calculated in proposed method. An average of the motion vectors can be achieved from both List\_0 and List\_1 in the HM reference software. We only use magnitudes of the MV directions for the x and y to calculate simply as shown in Eq. 4.

$$\text{MV}_{\text{abs}} = 0.5 \times \left( \sum_{i=0}^1 |\text{MV}_x(i)| + |\text{MV}_y(i)| \right). \tag{4}$$

To make use of the above equation, we have assume that the CU which has low motion should have a high chance of non-splitting. Since  $\text{MV}_{\text{abs}}$  is directly related with the motion of CU, we can infer that the CU which has low  $\text{MV}_{\text{abs}}$  value has a high chance that it will not split in the current hierarchy of CTB. We have checked the percentage of CUs which are not split for different values of  $\text{MV}_{\text{abs}}$ . The experimental result is shown in Table 4. We can observe that for  $\text{MV}_{\text{abs}} = 0$ , there is a quite high chance that the corresponding CU will not split. Using the Eq. (4), we can know the motion of CU with direct relationship with  $\text{MV}_{\text{abs}}$ . The CU which has low  $\text{MV}_{\text{abs}}$  value has high change that will not be divided in the current hierarchy of CU. Since the probability values of  $\text{avg}_d^{wf}$  and  $\text{MV}_{\text{abs}}$  are not so high (65–75 %), we cannot take any straightforward decision. Therefore, we need to combine this with other features in our main algorithm.

According to the above-mentioned method, the proposed algorithm can determine which CU needs to split early. The CU decision part in our algorithm can be divided into two stages. The first stage is to consider the CTU case. In terms of the CTU, we do not have any information from higher levels. Hence, ratio function cannot be calculated. In the second stage, other higher level CUs are considered. Figure 2 shows the flowchart of CU splitting termination (CST) algorithm. This proposed CST method checked the

**Table 1** Classified weight factors for all prediction modes according to the motion activities

Mode	Motion activity	PU_WF ( $\gamma_i$ )
SKIP	Homogeneous region with motionless	0
Inter 2 Nx2 N	Slow motion	1
Inter 2NxN, Nx2N	Motion between slow or moderate	2
Other inter and intra	Complex motion or texture	3

**Table 2** Results of the percentage of CUs which are not split in the next level

Sequence	CUs which are not split (%)			
	PU_WF = 0	PU_WF = 1	PU_WF = 2	PU_WF = 3
Nebula (A)	97.15	63.21	45.95	18.93
Park Scene (B)	91.35	58.98	42.36	16.68
Party Scene (C)	95.16	62.33	43.24	15.29
BlowingBubbles (D)	93.51	61.32	41.68	11.81
Average	94.29	61.46	43.30	15.67

**Table 3** Results of the percentage of CUs which are not split in the next level if their RD cost  $t < \text{avg}_d^{\text{wf}}$ 

Sequences	CU size	PU_WF			
		0	1	2	3
Nebula (A)	64 × 64	72.56	72.96	68.21	2.63
	32 × 32	73.24	71.36	67.36	7.36
	16 × 16	78.24	72.36	68.65	5.32
Park Scene (B)	64 × 64	72.35	69.36	66.32	6.35
	32 × 32	69.35	68.34	64.23	7.23
	16 × 16	68.23	67.29	66.37	8.28
Party Scene (C)	64 × 64	78.25	73.25	68.48	2.21
	32 × 32	76.35	72.39	71.24	1.58
	16 × 16	77.84	71.54	72.45	4.23
BlowingBubbles (D)	64 × 64	69.32	73.21	71.54	2.23
	32 × 32	71.25	71.23	71.11	5.41
	16 × 16	72.23	72.84	69.18	2.71

**Table 4** Results of the percentage of CUs which are not split in the next level for different  $MV_{\text{abs}}$  values

Sequence	CUs which are not split (%)			
	$MV_{\text{abs}} = 0$	$1 \leq MV_{\text{abs}} \leq 2$	$3 \leq MV_{\text{abs}} \leq 5$	$MV_{\text{abs}} > 5$
Nebula (A)	76.15	4.36	2.14	1.22
Park Scene (B)	72.96	3.51	3.26	1.96
Party Scene (C)	71.53	2.79	2.12	2.13
BlowingBubbles (D)	68.73	3.92	2.73	1.27
Average	72.34	3.64	2.56	1.64

PU mode weighting factor initially. If PU\_WF is 0 meaning SKIP mode, the decision is taken directly that there is no need of splitting. If PU\_WF is 1 or 2, then we cannot make any decision directly. Hence, these methods are divided into two stages to use different checking. For the CTU case, this algorithm only use  $MV_{\text{abs}}$  and local average RD cost of equivalent dimension of PU for PU\_WF = 1 and 2. If the RD cost of the current PU is less than average

RD cost in same dimension and  $MV_{\text{abs}}$  is 0, encoder is selected such that there is no need of splitting in the CTU case for PU\_WF = 1 and 2. Otherwise, ratio function,  $MV_{\text{abs}}$ , and local average RD cost of suitable dimension of PU for PU\_WF = 1 and 2 are used in non-CTU case. A non-CTU case is of same process. Since it can use the ratio function with higher level's information, if the RD cost is less than average RD cost of suitable dimension of PU and ratio function (R) is less than 0.25, it can also decide that CU is no need to split.

## 2.2 Fast prediction unit (FPU) decision technique

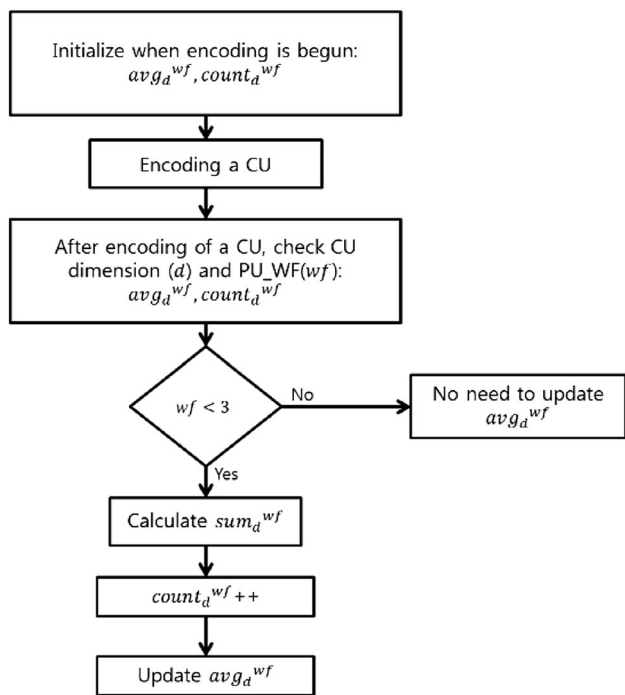
In this subsection, we propose an effective prediction unit selection method based on correlation and block motion complexity (BMC) that can be applied together with the CST above proposed. A natural video sequence has high spatial and temporal correlations. To analysis these correlations, we define predictor set ( $\Omega$ ) such as PU<sub>1</sub> to PU<sub>7</sub>. Figure 3 shows position from current CU for each PU<sub>*i*</sub>.

PU<sub>*i*</sub> is set as the mode of PU<sub>1</sub> to PU<sub>7</sub> from PU of current CU. PU<sub>1</sub>, PU<sub>2</sub>, PU<sub>3</sub>, PU<sub>6</sub>, and PU<sub>7</sub> are spatial PUs. PU<sub>4</sub> and PU<sub>5</sub> are temporal PUs. PU<sub>1</sub>, PU<sub>2</sub>, and PU<sub>3</sub> are PUs at above-left, above, and left PU from current CU, respectively. PU<sub>4</sub> is the co-located PU at forward direction and PU<sub>5</sub> is the co-located PU at backward direction. PU<sub>6</sub> and PU<sub>7</sub> are co-located parent PUs in depth-1 and depth-2, respectively.

We have explored correlation for predictor (PU<sub>*i*</sub>) in environment of 50 frames, 4 sequences of Class D, and random access with main profile configuration. Table 5 shows the probability of  $P(\text{Mode}_{\text{current}} = \text{PU}_i | \text{PU}_i, i = 1, 2, 3, \dots, 7)$ . It is clear that temporal and depth correlation values are relatively small, while correlation of PU<sub>1</sub>, PU<sub>2</sub>, PU<sub>3</sub> is very large.

To more investigate for temporal correlation for PU<sub>4</sub> and PU<sub>5</sub>, we have considered structure of temporal distance of the reference picture. Figure 4 indicates the structure of random access configuration in case of basic size of group of picture (GOP) is 8. In this structure, frame has distance with reference frame according to reference position and it can be represented as temporal Level<sub>*i*</sub>. Temporal Level<sub>1</sub> is equal to 8 from the reference frame. The distance of Level<sub>2</sub>, Level<sub>3</sub>, and Level<sub>4</sub> are 4, 2, and 1 according to temporal decomposition, respectively.

Table 6 shows the probability that the mode of the current CU is the same with the mode of the co-located CU in forward or backward reference frame. This result has considered temporal distance of the reference frame from the current frame. The temporal distance is represented by letter *d*. If temporal distance is smaller, then it has larger probability that a PU is encoded with the same mode with mode of the reference PU.



**Fig. 2** Flowchart of local average of RD cost calculation of CST algorithm

From the defined and observed factors, we can deduce a new block motion complexity model. To define block motion complexity, we have designed weight factor according to location from current CU as shown in Table 7. Table 7 indicates to assign more weight in spatial correlation and upper level (depth ‘d – 1’) from current depth

‘d’ than other position. Based on the designed mode weight factor (Table 1) for motion activities, we defined a block motion complexity (BMC) [28] according to the predefined mode weight factors ( $\gamma$ ) and a group of predictors in  $\Omega$  to estimate the PU mode of the current CU.

The BMC has been represented as follows: [28]:

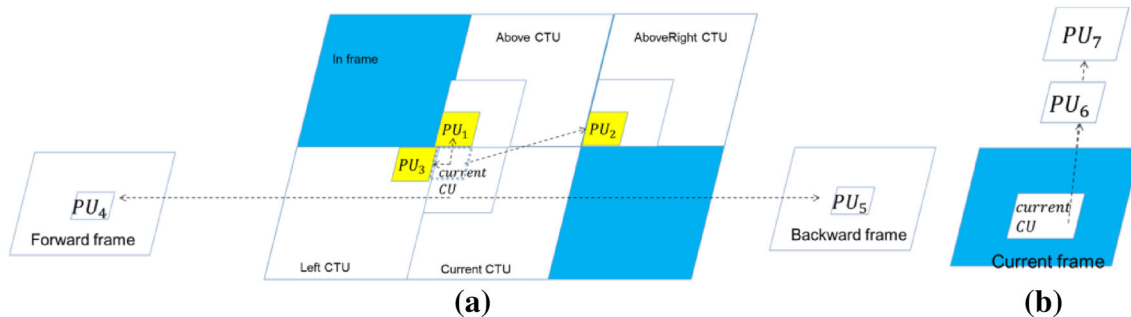
$$BMC = \frac{\sum_{i=1}^N W(i, l) \cdot k_i \cdot r_i}{\sum_{i=1}^N W(i, l) \cdot k_i}, \tag{5}$$

$$W(i, l) = w_i + f(i, l), \tag{6}$$

$$f(i, l) = \text{round}(l \cdot T_{\text{level}}) \cdot T_i \cdot Tk_i \tag{7}$$

where  $N$  is the number of PUs equal to 7,  $W(i, l)$  is function of the weight factor for each mode.  $\gamma_i$  is a weight factor for mode of  $PU_i$  in  $\Omega$ . The  $k_i$  denotes the available PUs. If the current CTB is in boundary region in the frame, then above or left PU information is not available. When  $PU_i$  is available,  $k_i$  is set to 1; otherwise,  $k_i$  is equal to 0.

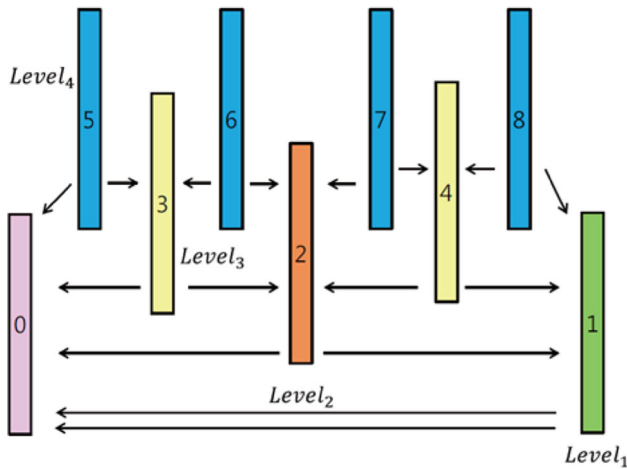
Equation (6) is an adaptive weighting factor function. The value of  $w_i$  is weight factor for  $PU_i$ . It has property of  $\sum_{i=1}^N w_i = 1$ . A value of  $l$  denotes the temporal decomposition layer. The HEVC consists of three layers in 8 GOP according to distance between frames. Accordingly, temporal layer can represent Level<sub>1,2,3,4</sub> having 8, 4, 2, and 1 of distance. If the current frame is close with reference frame, it has strong correlation than long-distance case. Therefore, setting weight factors is changed when PU has 1 and 2 distances adaptively as second row in Table 7.  $T_{\text{level}}$  is set to 0.02 and is used for normalization of temporal level as  $l$ .  $T_k$  is a control value for changing the weight factors.  $Tk_i$  is a control value used to increase or decrease weight factors.



**Fig. 3** The position of adjacent PUs from current CU for (a) spatio-temporal and (b) depth correlation

**Table 5** The probability of spatial, temporal, and depth correlation for adjacent PUs from current CU ( $PU_1$  to  $PU_7$  are located in Fig. 3)

Seqs.	$PU_1$ (%)	$PU_2$ (%)	$PU_3$ (%)	$PU_4$ (%)	$PU_5$ (%)	$PU_6$ (%)	$PU_7$ (%)
BasketballPass	88.35	89.44	89.46	58.70	59.57	84.08	68.95
BlowingBubbles	71.86	74.75	75.64	23.67	24.18	64.94	43.89
BQSquare	86.38	86.96	87.50	43.53	45.14	80.26	68.61
Racehorses	62.00	64.20	63.58	17.86	18.60	48.99	23.87
Total average	79.72	81.69	81.81	37.58	38.69	73.35	56.27



**Fig. 4** Coding order and level of temporal layer for each of the frames in random access configuration (size of the GOP = 8)

**Table 6** The probability of correlation of the predicted mode according to the temporal distance

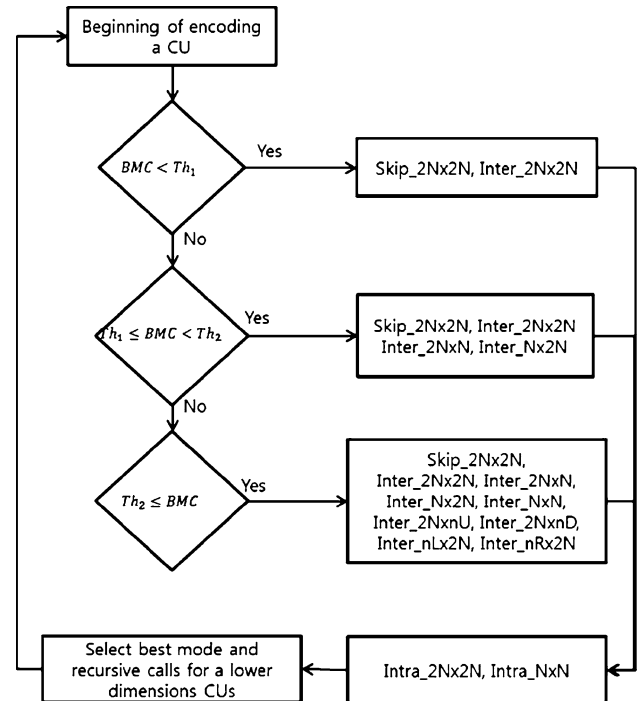
Sequence	QP	Level <sub>1</sub> <i>d</i> = 8 (%)	Level <sub>2</sub> <i>d</i> = 4 (%)	Level <sub>3</sub> <i>d</i> = 2 (%)	Level <sub>4</sub> <i>d</i> = 1 (%)	
Class D	BasketballPass	22	34.11	31.66	48.66	56.81
		27	39.28	41.22	56.45	65.51
		32	48.20	49.12	62.14	70.37
		37	52.78	56.29	66.89	72.80
BQ square		22	21.73	22.98	30.95	33.76
		27	23.94	25.20	34.80	52.12
		32	27.26	33.71	50.97	61.56
		37	27.29	32.66	55.38	68.68
	Total average		34.32	36.60	50.78	60.20

**Table 7** Weight factor for PUs (default setting and changing weight factor adaptively according to level of temporal layer)

Default setting	Index( <i>i</i> ) in classified PUs	1	2	3	4	5	6	7
	<i>w<sub>i</sub></i>	0.2	0.2	0.2	0.05	0.05	0.2	0.1
Changes of weight factor for PUs	Index( <i>i</i> ) in classified PUs	1	2	3	4	5	6	7
	<i>w<sub>i</sub></i>	0.2	0.2	0.2	0.1	0.1	0.1	0.1

It is set to 1, −2, or 0. PUs having the temporal correlations with PU<sub>4</sub> and PU<sub>5</sub> are set to be 1. Also, for a PU<sub>6</sub>, it is set to be −2. The others are set to 0.

$$\begin{cases} BMC < Th_1, & \text{Slow motion,} \\ Th_1 \leq BMC < Th_2, & \text{Medium motion,} \\ Th_2 \leq BMC, & \text{Complex motion or texture,} \end{cases} \quad (8)$$



**Fig. 5** Flowchart for fast prediction unit decision algorithm

where  $Th_1$  and  $Th_2$  are set to 1 and 3 as the mode weight factor for slow motion and complex motion or texture in PU\_WF in Table 1.

Figure 5 shows the flowchart for fast prediction unit decision algorithm. Before each PU is calculated, BMC is calculated and compared with each threshold. If the motion is slow, the proposed method performs SKIP mode and Inter 2 Nx2N. In case of medium motion, mode decision process performs modes of slow motion, Inter 2NxN and Nx2N. When motion is complex, all modes are performed, then mode search process selects the best one.

### 3 Experimental results

The proposed algorithm was implemented on the original HM 10.0 (HEVC reference software). Test conditions were random accessed using RA-Main. Standard sequences with 50 frames were used from two sequences per each class with various QP values (22, 27, 32, 37). Video sequences are classified as Class A to D sequences. Each class has different resolutions: Class A sequences—2560 × 1600 pixels, Class B sequences—1920 × 1080 pixels, Class C sequences—832 × 480 pixels, Class D sequences—416 × 240 pixels. Details of the encoding environment can be seen in JCTVC-L1100 [29].

To evaluate the performance, we defined the measurement of  $\Delta Bit$ ,  $\Delta PSNR_Y$ , and  $\Delta Time$  as:

**Table 8** The performance of the original HM reference software 10.0 and the proposed algorithm

Measurements	QP			CST			FPU [28]			MVM [25]			Proposed				
	$\Delta$ Bit (%)	$\Delta$ PSNR <sub>y</sub> (dB)	$\Delta$ Time (%)	BDBR	$\Delta$ Bit (%)	$\Delta$ PSNR <sub>y</sub> (dB)	$\Delta$ Time (%)	BDBR	$\Delta$ Bit (%)	$\Delta$ PSNR <sub>y</sub> (dB)	$\Delta$ Time (%)	BDBR	$\Delta$ Bit (%)	$\Delta$ PSNR <sub>y</sub> (dB)	$\Delta$ Time (%)		
Class A																	
Traffic																	
	22	0.15	-0.04	-19.80	0.9	0.62	-0.02	-41.47	1.2	0.55	-0.01	-19.96	0.9	1.02	-0.03	-52.77	1.9
	27	0.32	-0.03	-19.14		0.47	-0.02	-45.96		0.49	-0.02	-22.29		1.01	-0.03	-52.84	
	32	0.15	-0.02	-15.00		0.25	-0.04	-48.16		0.27	-0.02	-22.23		0.66	-0.04	-53.03	
	37	0.09	-0.02	-10.63		0.18	-0.04	-50.29		0.09	-0.02	-22.55		0.66	-0.04	-56.12	
	Avg	0.18	-0.03	-16.14		0.38	-0.03	-46.47		0.35	-0.02	-21.76		0.84	-0.04	-53.69	
PeopleOnstreet																	
	22	0.89	-0.04	-14.90	1.7	0.80	-0.01	-24.51	1.5	1.09	-0.01	-22.00	2.0	1.60	-0.02	-32.03	2.2
	27	0.76	-0.05	-15.05		0.86	-0.02	-29.10		1.36	-0.03	-24.66		1.41	-0.04	-35.79	
	32	0.55	-0.05	-15.75		0.83	-0.04	-32.54		1.30	-0.04	-25.63		1.09	-0.05	-42.75	
	37	0.33	-0.04	-16.82		0.64	-0.05	-35.24		1.09	-0.05	-25.25		1.07	-0.06	-45.47	
	Avg	0.63	-0.04	-15.63		0.78	-0.03	-30.35		1.21	-0.03	-24.39		1.29	-0.04	-39.01	
Avg of class A																	
		0.41	-0.04	-15.89	1.3	0.58	-0.03	-38.41	1.35	0.78	-0.03	-23.07	1.45	1.07	-0.04	-46.35	2.05
Class B																	
Cactus																	
	22	0.07	-0.04	-18.84	1.2	0.18	-0.01	-30.96	1.2	0.45	-0.02	-17.73	1.6	-0.11	-0.02	-41.82	2.3
	27	0.49	-0.02	-17.23		0.51	-0.02	-38.79		0.72	-0.02	-20.74		1.14	-0.03	-46.71	
	32	0.27	-0.02	-13.71		0.32	-0.02	-41.21		0.46	-0.03	-20.25		1.16	-0.03	-48.07	
	37	0.18	-0.02	-14.42		0.37	-0.03	-45.45		0.46	-0.03	-21.93		0.92	-0.05	-49.76	
	Avg	0.25	-0.02	-16.05		0.35	-0.02	-39.10		0.52	-0.03	-20.16		0.78	-0.03	-46.59	
BQTerrace																	
	22	-0.69	-0.06	-19.81	1	0.40	-0.01	-43.99	1.5	0.15	-0.01	-19.18	1	-0.08	-0.04	-54.58	1.5
	27	-1.11	-0.04	-16.34		0.45	-0.02	-50.47		0.27	-0.01	-19.90		-0.11	-0.05	-58.33	
	32	-0.90	-0.03	-15.73		0.38	-0.02	-54.12		0.39	-0.01	-21.36		-0.02	-0.05	-60.29	
	37	-0.27	-0.01	-9.05		0.11	-0.02	-56.40		0.45	0.00	-20.49		0.08	-0.06	-61.01	
	Avg	-0.74	-0.03	-15.23		0.33	-0.02	-51.24		0.31	-0.01	-20.23		-0.03	-0.05	-58.55	
Avg of class B																	
		-0.25	-0.03	-15.64	1.1	0.34	-0.02	-45.17	1.35	0.42	-0.02	-20.20	1.3	0.37	-0.04	-52.57	1.9
Class C																	
BasketballDrill																	
	22	0.82	-0.03	-25.80	1.1	1.01	-0.03	-39.17	1.6	0.90	-0.04	-23.73	1.3	1.34	-0.04	-53.60	2.3
	27	0.39	-0.03	-26.29		0.56	-0.05	-42.07		0.57	-0.03	-24.59		0.77	-0.06	-56.38	
	32	0.56	-0.02	-26.86		0.33	-0.04	-43.98		0.37	-0.03	-25.51		1.28	-0.05	-58.04	
	37	0.48	-0.01	-27.57		0.45	-0.04	-46.18		0.09	-0.04	-25.81		1.13	-0.06	-59.02	
	Avg	0.56	-0.02	-26.63		0.59	-0.04	-42.85		0.48	-0.03	-24.91		1.13	-0.05	-56.76	
PartyScene																	
	22	1.06	-0.09	-25.05	2.3	0.96	-0.02	-34.88	1.5	0.58	-0.02	-24.14	1	1.44	-0.09	-49.53	3.1
	27	0.84	-0.09	-24.55		0.83	-0.04	-38.27		0.43	-0.03	-26.61		1.16	-0.09	-50.82	
	32	0.32	-0.07	-24.53		0.32	-0.05	-40.67		0.46	-0.03	-27.21		1.26	-0.08	-53.39	
	37	0.35	-0.06	-24.83		0.33	-0.04	-43.96		0.56	-0.02	-27.14		0.97	-0.08	-56.41	
	Avg	0.64	-0.08	-24.74		0.61	-0.04	-39.45		0.51	-0.02	-26.27		1.21	-0.09	-52.53	

Table 8 continued

Measurements	QP	CST				FPU [28]				MVM [25]				Proposed			
		ΔBit (%)	ΔPSNR <sub>Y</sub> (dB)	ΔTime (%)	BDBR	ΔBit (%)	ΔPSNR <sub>Y</sub> (dB)	ΔTime (%)	BDBR	ΔBit (%)	ΔPSNR <sub>Y</sub> (dB)	ΔTime (%)	BDBR	ΔBit (%)	ΔPSNR <sub>Y</sub> (dB)	ΔTime (%)	BDBR
Avg of class C		0.6	-0.05	-25.69	1.7	0.6	-0.04	-41.15	1.55	0.49	-0.03	-25.59	1.15	1.17	-0.07	-54.65	2.7
Class D																	
BasketballPass	22	1.07	-0.06	-16.72	1.7	1.68	-0.06	-41.68	2.4	1.06	-0.03	-23.92	1.6	2.30	-0.11	-53.25	4.7
	27	0.38	-0.05	-14.80		0.82	-0.04	-44.40		0.92	-0.02	-24.82		2.21	-0.10	-56.82	
	32	1.09	-0.05	-15.22		1.96	-0.05	-47.22		1.21	-0.03	-24.59		3.41	-0.09	-58.32	
	37	-0.14	-0.03	-13.47		-0.12	-0.10	-50.54		0.65	-0.03	-23.59		0.76	-0.13	-59.52	
Avg		0.6	-0.05	-15.05		1.08	-0.06	-45.96		0.96	-0.03	-24.23		2.17	-0.01	-56.98	
BlowingBubbles	22	0.52	-0.05	-13.85	1.6	1.03	-0.04	-31.70	2.3	0.94	-0.02	-21.63	1.4	1.42	-0.09	-43.05	3.2
	27	0.53	-0.06	-17.59		1.15	-0.05	-35.17		0.58	-0.03	-22.30		1.97	-0.07	-48.49	
	32	-0.07	-0.04	-19.22		1.03	-0.06	-38.31		0.71	-0.03	-22.01		1.61	-0.06	-52.90	
	37	-0.12	-0.07	-20.57		0.38	-0.06	-41.32		0.86	-0.03	-21.64		0.54	-0.04	-54.42	
Avg		0.22	-0.06	-17.81		0.90	-0.05	-36.62		0.77	-0.03	-21.90		1.39	-0.06	-49.71	
Avg of class D		0.41	-0.05	-16.43	1.65	0.99	-0.06	-41.29	2.35	0.87	-0.03	-23.06	1.5	1.78	-0.08	-53.35	3.95
Total average		0.29	-0.04	-18.41	1.44	0.63	-0.04	-41.51	1.65	0.64	-0.03	-22.98	1.35	1.09	-0.05	-51.72	2.65

$$\Delta\text{Bit} = \text{Bit}_{\text{proposed}} - \text{Bit}_{\text{original}} \tag{9}$$

$$\Delta\text{PSNR}_Y = \text{PSNR}_{Y_{\text{proposed}}} - \text{PSNR}_{Y_{\text{original}}} \tag{10}$$

$$\Delta\text{Time} = \frac{\text{Time}_{\text{Proposed}} - \text{Time}_{\text{Anchor}}}{\text{Time}_{\text{Anchor}}} \times 100 \tag{11}$$

ΔTime is a complexity comparison factor used to indicate the amount of total encoding time saving (Eq. (11)). From Eq. (11), Time<sub>(x)</sub> means the total consumed time of the method x for encoding.

Table 8 shows results for comparisons between the original HM 10.0 software and the proposed algorithm without any fast options. The Bjøntegaard Delta bitrates (BDBR) are shown in Table 8 [30]. Moreover, we have considered three other algorithms (namely CST, FPU and MVM) which are shown in Table 8. From Table 8, it can be observed that CST reduces encoding time from 9.05 to 27.57 %, 18.41 % on average with 1.44 % of the BDBR from 0.29 % and 0.04 dB losses both in bit rate and Y-PSNR, respectively. The FPU has achieved the computational complexity reduction from 24.51 to 56.40 %, 41.51 % on average. Meanwhile, the BDBR between FPU and HM 10.0 is 1.65 %, and bit rate and Y-PSNR losses are 0.63 % and 0.04 dB. Performance factor of time saving between MVM and HM 10.0 is 19.18 to 25.81 %, 22.98 % on average with 0.64 % of bit increment and 0.03 dB of quality loss. The BDBR of MVM is 1.35 % on average.

The proposed method reduces the computational complexity from 32.03 to 61.01 %, 51.72 % on average with some loss as 1.09 %, and 0.05 (dB) losses both in bit rate and Y-PSNR, respectively. The BDBR between proposed algorithm and HM 10.0 is 2.65 % on average. It can be noted from the table that the proposed algorithm gives superior result in terms of time reduction compared with other fast algorithms. However, it suffers from marginal loss of 2.65 % in BDBR including bit rate and PSNR losses. But if we consider the time reduction, then the loss is negligible.

The proposed algorithm increases the speed of the HEVC encoding system up to 61.01 % on BQTerrace sequence in Class B sequence, compared to the full mode search. Compared to the MVM method, the proposed algorithm achieved speed-up gain of up to 32.37 % with a smaller bit increment for Class B sequences. Using the proposed algorithm, we can see that the average speed-up gain of over 52 % was obtained comparing to the full mode search while keeping negligible bit rate increment. For Class A sequences, the average speed-up factor of 46.35 % was achieved by the proposed method with 1.07 % of bit increment and 2.05 % BD rate loss. When comparing to the MVM method, about 29 % of speed-up factor has been obtained.

In terms of bit rate, the proposed method achieved small increment of bit rate (0.32 %) in Class B sequences as



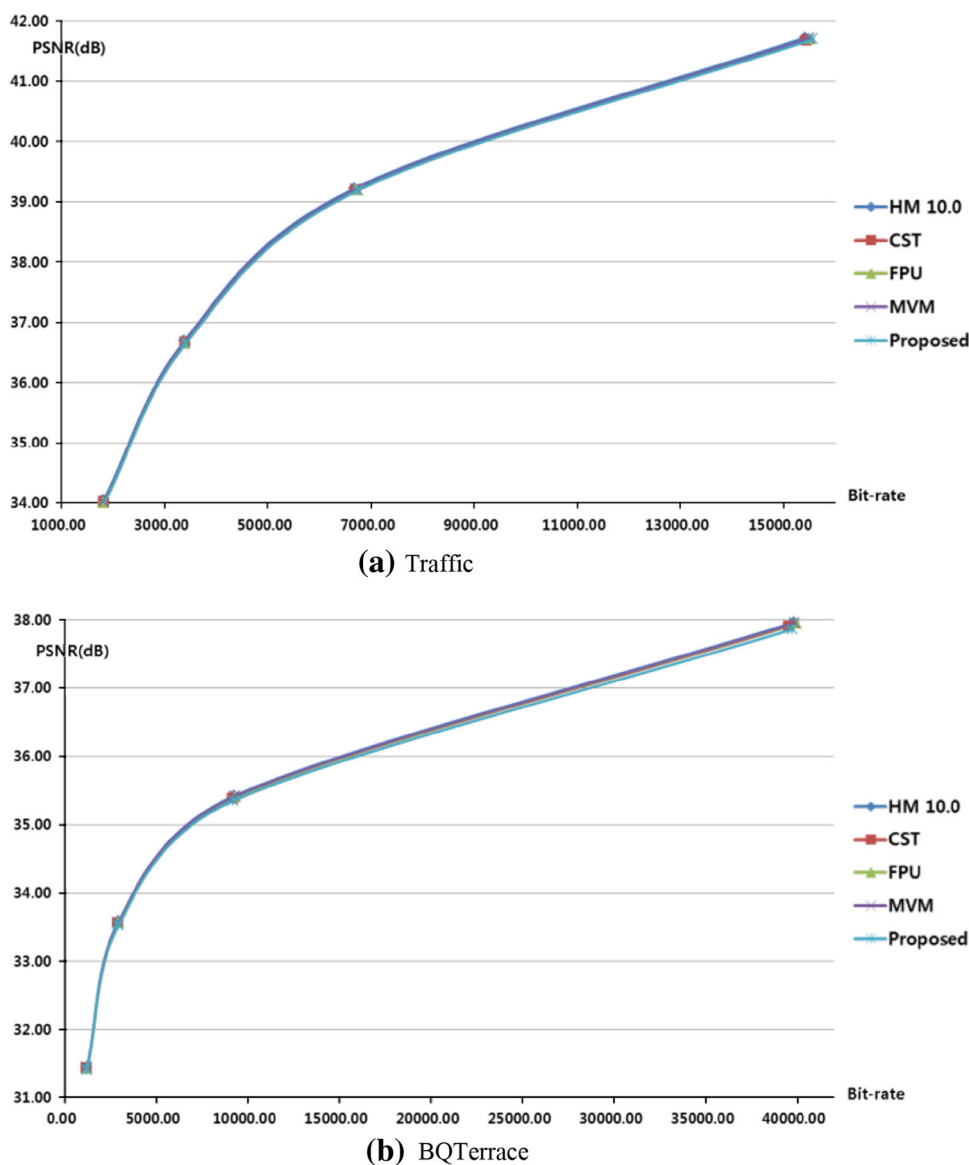
minimum bit increment. For Class D sequences, bit increment of up to 1.78 % was obtained. The proposed algorithm has superior result in terms of time reduction compared with other fast algorithms. Although, the proposed method has little more bit rate than other fast algorithms, the computational complexity was reduced almost 33, 28.74, and 10 % than the CST, MVM, and FPU, respectively. This means that the proposed algorithm can provide credible quality for larger video resolution, with high speed-up factor. From these results, the proposed algorithm is efficient to make real-time encoding system without significant degradation of the encoding performance.

Figures 6 and 7 show the RD performance of one of test sequences in Class A to D. In Fig. 6(a) and (b), the

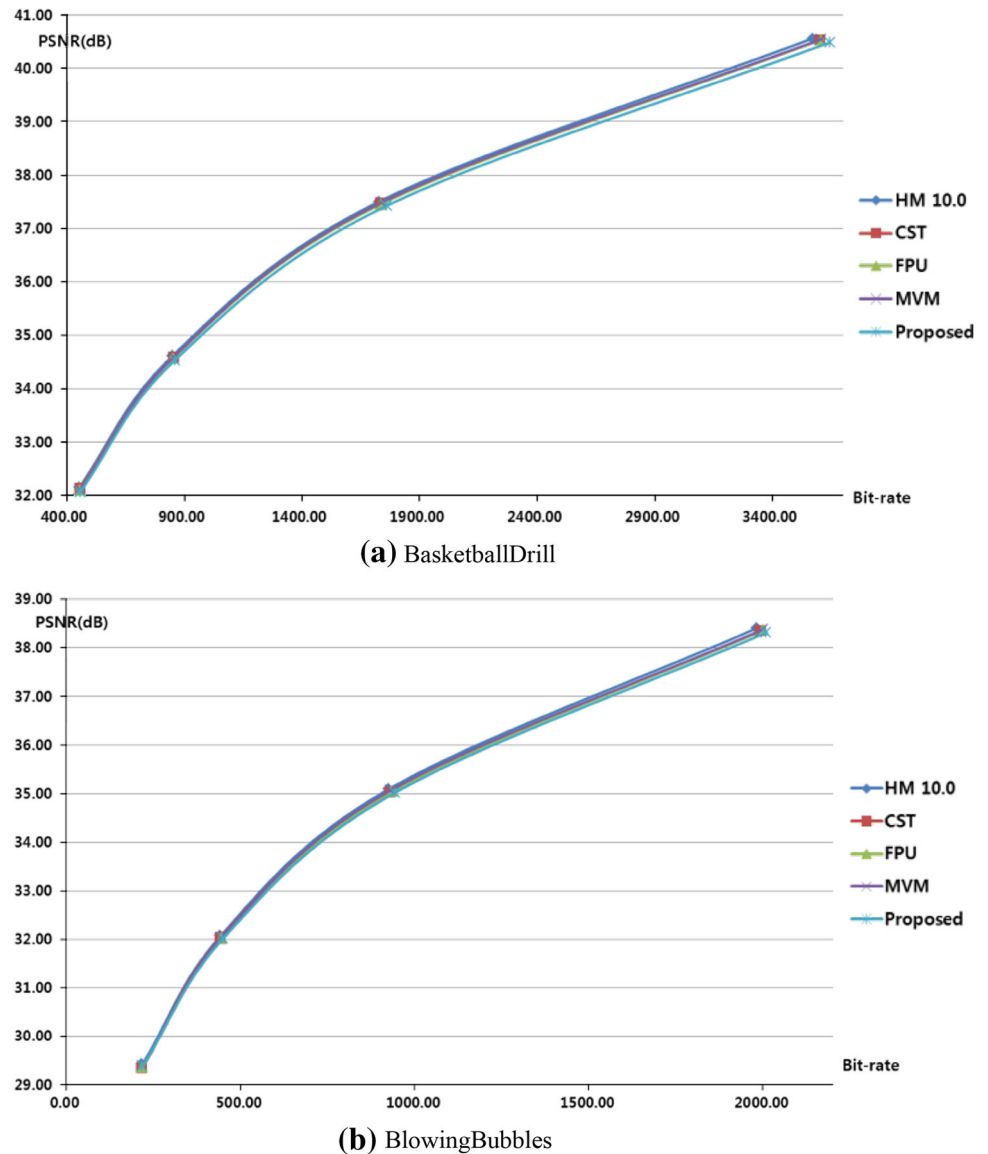
proposed method is very similar to the original HM 10.0 software. There is negligible loss of bit rate. However, it is shown that for the bigger QP values, larger loss are generated for bit rate in Fig. 7 (a) and (b). Although our algorithm has small loss for bit rate and PSNR, it has significant time-saving factor of 32.03 % at least and up to 61.01 %. In performance of Class B, the proposed algorithm can achieve very little loss as 0.37 % of bit rate and 0.04 (dB) of PSNR, and 52.57 % of time-saving performance without any fast options.

For the BQTerrace (Class B) sequence, there is negligible loss of quality. The proposed method gives almost similar performance to the original HM encoder. In the overall bit range, we can see that the proposed algorithm gives almost similar quality with same bit rate. This means

**Fig. 6** Rate–distortion (RD) curves for (a) Traffic (Class A) and (b) BQTerrace (Class B) sequences in random access, main profile condition



**Fig. 7** Rate–distortion (RD) curves for (a) BasketballDrill (Class C) and (b) BlowingBubbles (Class D) sequences in random access, main profile condition

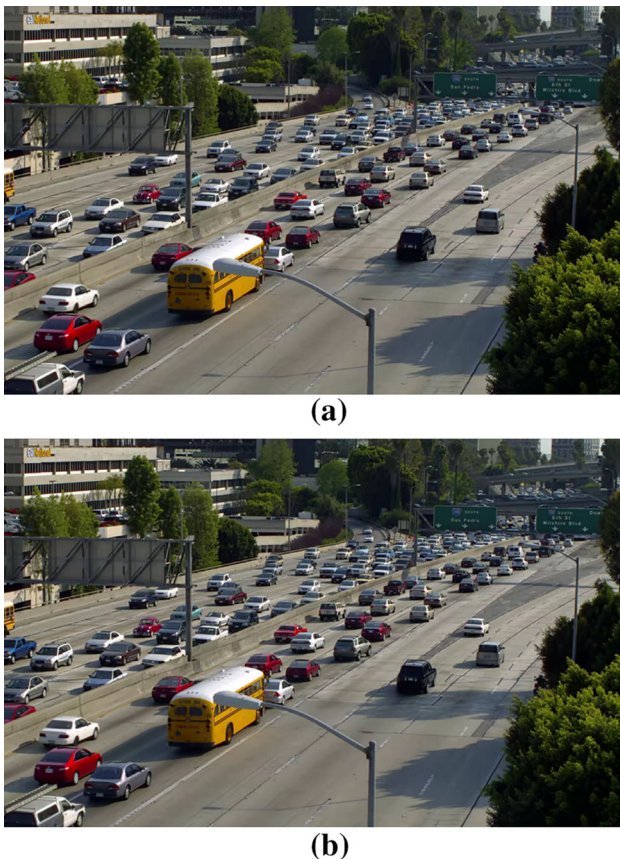


that the suggested algorithm can keep a reliable video quality with speeding up the HM encoder by about 53 %. Up to 20 Mbps, we can see that the performance of the proposed algorithm is credible in terms of bit rate and image quality. From 25 Mbps, the proposed algorithm begins to show a little loss of the quality.

In case of the BasketballDrill (Class C) sequence, the performance of the proposed algorithm is similar to the original HM encoder in terms of bit rate and image quality up to 15 Mbps. In the range of high bit rate (from 20 Mbps), a slight larger loss of the quality was shown by about 0.07 (dB) of average value. This means that the proposed algorithm can provide very high speed-up factor while the quality loss may become larger than that of larger size of video.

We can know this loss is negligible as shown in Figs. 6, 7, and 8. Figure 8 shows a subjective comparison with the 20th reconstructed image of the original HM 10.0 reference software (Fig. 8(a)) and the proposed algorithm (Fig. 8(b)). The reconstructed images are very similar. It is difficult to find the difference of the quality in visual. This means the quality loss of the proposed method is negligible.

The proposed scheme also has good structure that can be run with other fast options in HEVC standard. Therefore, the proposed algorithm can be a very helpful real-time system with intelligent user interface for interactive multimedia system that handling large graphic data such as image processing data, broadcasting media, cloud computing media, and web video data.



**Fig. 8** Comparison with the reconstructed image of the original HM 10.0 encoder and the proposed algorithm: Traffic Sequence (2560 x 1600, Class A, QP value is 37, the 20th frame). **(a)** Reconstructed image of the original HM 10.0 version. **(b)** Reconstructed image of the proposed algorithm

#### 4 Conclusions

We have proposed a new fast encoding algorithm both of early CU splitting termination (CST) and fast PU decision (FPU) for HEVC-based interactive multimedia system. The CST algorithm used the RD costs for different CU dimensions and motion complexity for PU level. The FPU method is based on spatial, temporal and depth correlation information and adaptive weighting factor design using temporal distance. By combining CST and FPU, the proposed algorithm achieved, on average, a 51.72 % of time saving over the original HM 10.0 software and BDBR is 2.65 % with 1.09 % of bit rate loss. The proposed algorithm is useful for designing a real-time HEVC video encoding system with intelligent user interface for interactive multimedia services.

**Acknowledgments** This work was supported by the ICT R&D program of MSIP/IITP. [13-912-02-002, Development of Cloud Computing Based Realistic Media Production Technology].

#### References

1. Maybury, M.T.: Intelligent user interfaces: an introduction. In: 4th international conference on intelligent user interfaces, 1999
2. Yo-Sung, H.: Challenging technical issues of 3D video processing. In: *J Converg (JoC)*, vol. 4, No. 1, pp. 1–6, Mar (2013)
3. Song, E.-H., Kim, H.-W., Jeong, Y.-S.: Visual monitoring system of multi-hosts behavior for trustworthiness with mobile cloud. In: *J Info Process Syst (JIPS)*, vol. 8, No. 2, pp. 347–358, Jun 2012
4. Tsai, J.C., Yen, N.Y.: Cloud-empowered multimedia service: an automatic video storytelling tool. In: *J Converg (JoC)*, vol. 4, No. 3, pp. 13–19, Sept (2013)
5. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. In: *IEEE transaction on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, Dec (2012)
6. Wiegand, T., Sullivan, G.J.: The H.264/AVC video coding standard. In: *Signal processing magazine, IEEE*, vol. II, pp. 148–153, 2007
7. Tan, H.L., Liu, F., Tan, Y.H., Yeo, C.: On fast coding tree block and mode decision for high-efficiency video coding (HEVC). In: *International Conference on Acoustics, Speech and Signal Processing, IEEE*, pp. 825–828, Mar (2012)
8. Kim, J., Jeong, S., Cho, S., Choi, J.S.: Adaptive coding unit early termination algorithm for HEVC. In: *International Conference on Consumer Electronics (ICCE)*, Las Vegas, Jan (2012)
9. Leng, J., Sun, L., Ikenaga, T., Sakaida, S.: Content based hierarchical fast coding unit decision algorithm for HEVC. In: *International Conference on Multimedia and Signal Processing, China*, May 2011
10. Shen, X., Yu, L., Chen, J.: Fast coding unit size selection for HEVC based on Bayesian Decision Rule. In: *Picture Coding Symposium (PCS)*, May (2012)
11. Shen, X., Yu, L.: CU splitting early termination based on weighted SVM. In: *EURASIP J Imag Video Process*, 2013
12. Kim, J., Yang, J., Won, K., Jeon, B.: Early determination of mode decision for HEVC. In: *Picture Coding Symposium*, pp. 449–452, May (2012)
13. Lee, J., Jeon, B.: Fast method decision for H.264. In: *IEEE International Conference Multimedia and Expo (ICME)*, June 2004
14. Choi, I., Lee, J., Jeon, B.: Fast coding mode selection with rate-distortion optimization for MPEG-4 Pare-10 AVC/H.264. In: *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557–1561, Dec (2006)
15. Zhang, H., Ma, Z.: Early termination schemes for fast intra mode decision in high efficiency video coding. In: *IEEE International Symposium on Circuits and Systems, ISCAS*, May (2013)
16. Jiang, W., Ma, H., Chen, Y.: Gradient based fast mode decision algorithm for intra prediction in HEVC. In: *IEEE International Conference on Consumer Electronics, Communications and Networks (CECNet)*, (2012)
17. Teng, S.-W., Hang, H.-M., Chen, Y.-F.: Fast mode decision algorithm for residual quadtree coding in HEVC. In: *Visual Communications and Image Processing, IEEE*, pp. 1–4, Nov 2011
18. Tian, G., Goto, S.: Content adaptive prediction unit size decision algorithm for HEVC intra coding. In: *Picture Coding Symposium (PCS)*, May (2012)
19. Wu, D., Pan, F., Lim, K.P., Wu, S., Li, Z.G., Lin, X., Rahardja, S., Ko, C.C.: Fast intermode decision in H.264/AVC video coding. In: *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 6, pp. 953–958, July (2005)
20. Kim, J.-H., Kim, B.-G., Cho, C.-S.: A fast mode decision algorithm using a block correlation in H.264/AVC. In: *International Symposium on Consumer Electronics, IEEE*, pp. 1–5, June (2011)
21. Choi, K., Jang, E.S.: Fast coding unit decision method based on coding tree pruning for high efficiency video coding. In: *Optical Engineering Letter*, Mar (2012)

22. Shen, L., Liu, Z., Zhang, X., Zhao, W., Zhang, Z.: An Effective CU Size Decision Method for HEVC Encoders. *IEEE Trans. Multimed.* **15**(2), 465–470 (2013)
23. Yang, S., Lee, H., Shim, H.J., Jeon, B.: Fast inter mode decision process for HEVC encoder. In: *Image, Video, and Multidimensional Signal Processing (IVMSP) Workshop*, (2013)
24. Ahn, S., Kim, M., Park, S.: Fast decision of CU partitioning based on SAO parameters, motion and PU/TU split information for HEVC. In: *Picture Coding Symposium (PCS)*, (2013)
25. Zeng, H., Cai, C., Ma, K.-K.: Fast mode decision for H.264/AVC based on macroblock motion activity. In: *IEEE Trans. on Circuits and Systems for Video technology*, pp. 1–11, vol. **19**, no. 4, Apr (2009)
26. Hosur, P.I., Ma, K.-K.: Motion vector field adaptive fast motion estimation. In: *Inter. Conference on Information, Communications and Signal Processing (ICICS'99)* (1999)
27. Hilmi, B., Goswami, K., Lee, J.-H., Kim, B.-G.: Fast inter-mode decision algorithm for H.264/AVC using macroblock correlation and motion complexity analysis. In: *International Conference on Consumer Electronics (ICCE)*, Las Vegas, Jan (2012)
28. Lee, J.-H., Park, C.-S., Kim, B.-G., Dong-San, J., Jung, S.-H., Choi, J.S.: Novel fast PU decision algorithm for the HEVC video standard. In: *IEEE International Conference on Image Processing*, pp. 1982–1985 (2013)
29. Bossen, F.: Common test conditions and software reference configurations. In: *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 12th Meeting*, Jan (2013)
30. Bjøntegaard, G.: Calculation of average PSNR differences between RD-curves. In: *ITU-T SG 16 Q.6 Document, VCEG-M33*, Austin, US, Apr (2001)



**Jong-Hyuk Lee** received his B.S. and M.S. degrees in the Department of Computer Engineering from SunMoon University, Korea, in 2010 and 2013, respectively. In 2013, he joined the Department of Computer Engineering for pursuing his Ph.D. degree at SunMoon university. His research interests include image signal processing algorithms, video coding techniques, and HEVC/SHVC video standard techniques.



**Kalyan Goswami** is Ph.D. Student at the Department of Computer Engineering, Sunmoon University, South Korea. He received the B.Tech. degree in Electronics and Telecommunication Engineering from Kalyani University, Calcutta, India, in 2004 and M.S. form IIT Kharagpur, India. Before joining IIT, he was working as a Programmer Analyst in Cognizant Technology Solutions. His research interests include algorithm development in the field

of Video Processing.



**Byung-Gyu Kim** has received his B.S. degree from Pusan National University, Korea, in 1996 and an M.S. degree from Korea Advanced Institute of Science and Technology (KAIST) in 1998. In 2004, he received a Ph.D. degree in the Department of Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST). In March 2004, he joined in the real-time multimedia research team at the

Electronics and Telecommunications Research Institute (ETRI), Korea where he was a senior researcher. In February 2009, he joined the Department of Computer Engineering at SunMoon University, Korea where he is currently a professor. In 2007, he served as an editorial board member of the *International Journal of Soft Computing*, *Recent Patents on Signal Processing*, *Research Journal of Information Technology*, *Journal of Convergence Information Technology*, and *Journal of Engineering and Applied Sciences*. He also served as Program Committee Member of CSIP 2011 and EMC 2012. He has published over 90 international journal and conference papers, patents in his field. His research interests include image and video object segmentation for the content-based image coding, video coding techniques, 3D video signal processing, wireless multimedia sensor network, embedded multimedia communication, and intelligent information system for image signal processing. He is a member of IEEE, ACM, and IEICE.



**Seyoon Jeong** received the B.S. and M.S. in electronics engineering from Inha University, Korea, in 1995 and 1997, respectively. Since 1996, he has been a senior member of research staff with ETRI, Korea, and he is also working toward the Ph.D. in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea. His current research interests include video coding, video transmission, UHDTV and

cloud computing.



**Jin Soo Choi** received the B.E., M.E., and Ph.D. in electronic engineering from Kyungpook National University, Korea, in 1990, 1992, and 1996, respectively. Since 1996, he has been a principal member of engineering staff in ETRI, Korea. He has been involved in developing the MPEG-4 codec system, data broadcasting system, and UDTV. His research interests include visual signal processing and interactive services in the field of the digital broadcasting

technology.