EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Efficient AMP decision and search range adjustment algorithm for HEVC

Xinpeng Huang[1,2], Ping An[1,2*] and Qiuwen Zhang[3]

Abstract

The advanced video encoder High Efficiency Video Coding (HEVC) utilizes several novel coding tools so that it can obtain improvement in coding performance for a huge number of video data. However, these tools increase the computational complexity greatly specially in the interprediction phase. Therefore, optimization for interprediction plays an important role in accelerating the whole HEVC encoding process. To this end, this paper proposes an efficient prediction algorithm for improving HEVC intercoding. According to the spatial correlation between the current block and its co-located block in the previous frame, we design a fast decision method so as to perform asymmetric partitions directly without the complicated calculation. Furthermore, based on the motion degree function we defined, the motion search range of motion estimation can be adaptively adjusted. Experimental results demonstrate that the proposed algorithm is able to save coding time significantly both under RandomAccess and Lowdelay P configurations, with a negligible drawback in term of Bjontegaard delta-rate increase.

Keywords: HEVC, Interprediction, Asymmetric partition, Search range

1 Introduction

High Efficiency Video Coding (HEVC), an advanced video coding standard, is a good substitute for encoding ultra-high-definition video. In contrast to other standards, HEVC employs several new coding tools to eliminate redundancy existing in inter- and intraframes [1–4]. The higher precision and more flexible partition structure is one of the major reasons that HEVC outperforms H.264/AVC [5]. Additionally, good performance of HEVC makes it be applied to several extensions [6–11]. In HEVC, every frame of video is divided into coding tree units (CTUs) and each CTU can be split recursively using quad-tree structure, until the minimum size is reached. Similar to macroblock concept in H.264/AVC, coding unit (CU) is the basic unit in HEVC that carries the whole coding information of current block [12, 13]. There are eight prediction units (PUs) shown in Fig. 1, and these eight PU partitions are named as PART_2N×2N, PART_N×N, PART_2N×N, PART_N×2N, PART_2N×nU, PART_2N×nD, PART_nL×2N, PART_nR×2N, respectively. Among these eight labels, uppercase N denotes half the length of current CU's length or width and lowercase n signals one quarter of that [14]. It is akin to CU that PU brings the information about prediction. If current CU is encoded with intraprediction, it is always divided using square partition, namely INTRA_2N×2N and INTRA_N×N. As illustrated in Fig. 1, a CU can either be coded as a single PU or it can be split into two or four rectangular PUs.

After PU division, motion estimation (ME) is performed to find out the optimal motion vector (MV) of current block through matching corresponding samples. The computational load of ME in HEVC is related to the motion search range in matching process. For the full search range, 128 candidate points (configured as [− 64, 64]) are adopted in horizontal and vertical directions, respectively. In other words, more than 16,000 points should be searched to find the best MV for current block. Although the fast ME approach TZ Search is added into the latest HEVC reference software to simplify the MV search process, the coding complexity is still very high. In interprediction stage, asymmetric partition (AMP) partitioning and ME process can promote the matching precision. However, the coding time is increased significantly for the novel complex design.

* Correspondence: anping@t.shu.edu.cn
[1] Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
[2] Key Laboratory of Ministry of Education for Advanced Display and System Application, Shanghai University, Shanghai 200072, China
Full list of author information is available at the end of the article

Therefore, this paper presents an efficient approach to reduce the calculation burden in interprediction. Firstly, we find out a fast decision method so that some prediction units are divided into asymmetric partitions directly without the complex calculation. And then, based on the motion degree function we defined, the motion search range of ME can be adaptively adjusted.

The remainder content of this article is organized as follows. Our proposed method is described in Section 2. Section 3 demonstrates the experimental comparison between our proposed approach and other methods and discusses the coding performance of the proposed algorithm. Finally, we summarize this paper in Section 4.

1.1 The related previous work

To reduce interprediction computational burden for HEVC encoder, many researchers have proposed various fast methods. These methods aim to simplify PU mode decision process that we sum up as follows. A different CU visiting order is stated in Ref. [15], and this modification reduces the complexity of CU size decision, mode decision, and interframe prediction. Another fast CU size prediction for HEVC is presented in Ref. [16], which early terminates partitioning modes to reduce computational complexity. Ref. [17] adopts convolution neural network to decrease no less than two CU partition modes for rate distortion optimization (RDO). A set of procedures for partition structure determination is designed in Ref. [18] to early terminate exhaustive RDO process using data mining techniques. An early terminating fast intermode decision algorithm, namely hierarchical intermode decision (HIMD), is presented in Ref. [19] using the correlation between RD cost and PU size. A course of low complexity mode decision algorithm consists of four early termination strategies and this approach uses co-located block depth information and current block size information to terminate RDO process [20]. A reverse CU access approach is proposed in Ref. [21] that can further reduce HEVC coding complexity, particularly for high motion activity sequences. A fast CU partitioning algorithm is provided in Ref. [22] for HEVC, and this method early terminates the CU size decision process based on the Bayesian decision rule. A fast interprediction mode decision algorithm is proposed in our previous work [23] that uses current block edge information to simplify RDO process so as to accelerate coding time for HEVC encoder.

Furthermore, complex ME process in HEVC encoder requires a huge calculation, so many low-complexity algorithms have been presented to make improvement on ME process. A block motion information derivation is described in Ref. [24] without conventional motion estimation and motion parameters. A fast algorithm is introduced in Ref. [25] to reduce the number of ME searching points from 81 to just 31. A fast ME decision method is provided in Ref. [26] that skips block search process adaptively on the basis of the relationship between the two types PU after classifying PUs into two classes. Two individual algorithms are proposed in Ref. [27] that obtain the best performance collaboratively. A better balance is stated between computational burden and RD performance in Ref. [28]. In Ref. [29], authors state fast mode decision algorithm not only for CU size decision but also for ME decision, and it achieves a great time saving.

Admittedly, all of these approaches have improved coding efficiency of intercoding for HEVC encoder. However, it is hard for interprediction optimization to balance the computational complexity and the coding performance. Motivated by the abovementioned issue, we propose an efficient AMP decision and search range adjustment algorithm for HEVC.

1.2 Preliminary work

Because of the huge amount of data, especially when having an advanced hierarchical partition design in current HEVC standard, compression of high-definition video content relies heavily on the efficient compression of video information. Note that, one way of performing prediction mode decision would exhaustively search the possibilities within a set containing all possible modes, which results in an undesirable complexity. To determinate the optimal mode, it adopts Lagrange multiplier to find the certain mode with the least RD cost as the best one, and the RD cost function (J) is shown as follows:

$$J = D_{\mathrm{mode}} + \lambda \times BR_{\mathrm{mode}} \tag{1}$$

where D_{mode} denotes the distortion between the current mode and the matching mode on average, λ signifies the Lagrange multiplier, and BR_{mode} is bits required for signaling the coding mode and the associated side information. Moreover, CU division way in HEVC is identical to that way in H.264/AVC, while new PU partitions (including square partition (SP), symmetric partition (SMP), and AMP) have been developed for HEVC interprediction. In contrast to H.264/AVC intermode prediction, before the decision of the optimal MV for a block, all the SP, SMP, and AMP in PU modes are added to the pre-decision list. And then, all possible search point candidates corresponding to the PU mode in pre-decision list are considered as full RD search list to perform ME using full search range. During this procedure, the distortion among all candidates (including PU partitions and search points) is computed and the best prediction mode is selected. This technique enhances the highest possible coding precision in HEVC encoder, but the “try all and select the best” method leads to

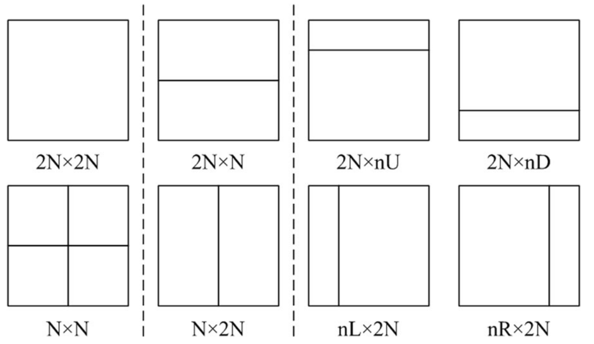


Fig. 1 The eight PU partitions. Left: square partition (SP); middle: symmetric partition (SMP); right: asymmetric partition (AMP)

extremely high coding complexity, so it limits HEVC for practical applications.

In preliminary work, prediction modes distribution has been checked at RA (RandomAccess configuration) and LDP (Lowdelay P configuration) conditions. Figure 2 shows the average area shares of prediction modes in HEVC encoder at RA and LDP conditions. It is obvious that with the reduction of quantization parameter (QP), more and more blocks choose AMP as the best interprediction mode under both RA and LDP configurations. Although the share of AMP is lower than other mode decision, it is very time-consuming to perform AMP for interprediction modes. Furthermore, rather than intracoding, all of interprediction modes need to be addressed with ME, which results in both high calculation burden and a long coding time. To analyze the computational load of AMP partitioning and ME process in HEVC encoder, extensive experiments were conducted on ten text sequences with QP setting as 22, 27, 32, and 37. These settings are tabulated in Table 1 according to common test condition (CTC) [30], and HEVC test model 14.0 (HM 14.0) is used as the experimental platform.

The encoder configuration is as follows: GOP length is set to 8 with an intraperiod being 32; the maximum CU size is 64 × 64, and CU depth level is set to 4; search range of ME is configured as [− 64, 64]. We execute simulated experiments twice with AMP disabled (off) and enabled (on) for each test sequence, and the original HM reference software (AMP on) is regarded as the benchmark. Moreover, computational load for these experiments is reflected by coding time which is calculated by $\Delta T = \frac{T_{\mathrm{off}}^{\mathrm{avg}} - T_{\mathrm{on}}^{\mathrm{avg}}}{T_{\mathrm{on}}^{\mathrm{avg}}}$, where $T_{\mathrm{off}}^{\mathrm{avg}}$ and $T_{\mathrm{on}}^{\mathrm{avg}}$ mean the average coding time when AMP off and AMP on, respectively. In addition, peak signal-to-noise ratio (PSNR) change, bitrate variation, and coding time change are measured by "BD-PSNR (dB)" (Bjontegaard Delta PSNR), "BD-BR (%)" (Bjontegaard Delta Bitrate), and "ΔT (%)" [31].

Table 2 illustrates the coding efficiency comparison between the HM 14.0 with AMP off and the original HM 14.0 with AMP on both under RA and LDP cases. The minus and the positive number respectively denote drop and augment compared with the data derived by the benchmark (the same as in the following tables). From the experimental results, one can notice from Table 1 that the maximum time saving of 18.12% for "BasketballDrive" under LDP configuration when AMP is disabled. The coding time is averagely reduced about 12.83 and 14.05% under "RA" and "LDP" case, respectively. It can be seen that disabling AMP can speed up encoding process. While for the sequences with small movement activity, such as "BQSquare" and "Johnny", PSNR is close to that of original HEVC encoder with AMP off. However, when AMP is disabled, both PSNR loss and bitrate increase are obviously significant. It also can be seen from Table 2 that for test sequences with much high motion activity areas, such as "BasketballDrive," "BasketballDrill," and "RaceHorses," the

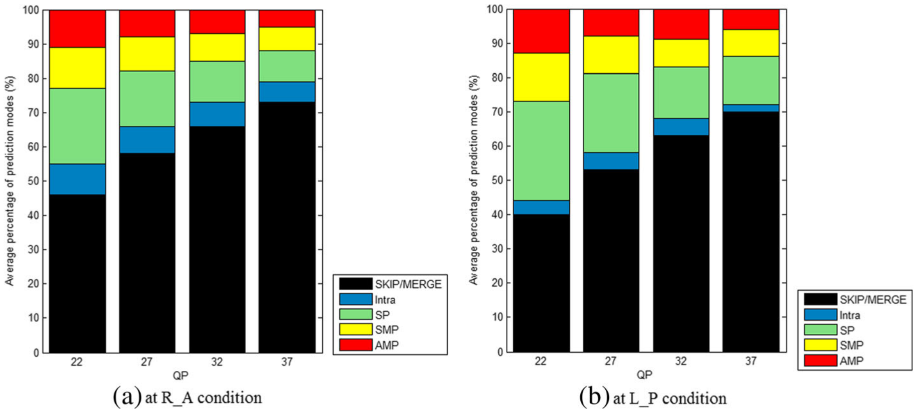


Fig. 2 Prediction mode distribution. **a** At R_A condition. **b** At L_P condition

Table 1 Test sequence information

Class	Resolution	Test sequence	Frame rate	Bit depth	Motion degree
A	2560 × 1600	Traffic	30 fps	8	High
		PeopleOnStreet	30 fps	8	Low
B	1920 × 1080	BasketballDrive	50 fps	8	High
		Cactus	50 fps	8	Low
C	832 × 480	BQMall	60 fps	8	Low
		BasketballDrill	50 fps	8	High
D	416 × 240	BQSquare	60 fps	8	Low
		RaceHorses	30 fps	8	High
E	1280 × 720	FourPeople	60 fps	8	Low
		Johnny	60 fps	8	Low

coding time reduction for them is relatively more than that of other test sequences after disabling AMP. Because turning off AMP can eliminate amounts of complicated ME process, computational load is saved conspicuously, and yet PSNR loss and bitrate increase are evident.

These results demonstrate that there is much room to simplify the interdecision process in HEVC. Accordingly, if we can exploit certain information to adaptively stop some specific blocks executing AMP prediction, and if the complex ME process can be simplified in HEVC encoder, the calculation load will be reduced dramatically in the whole HEVC interencoding process.

2 The proposed efficient method for HEVC intercoding

As presented in the previous sections, the complicated intermode decision consumes plenty of coding time, which prevents HEVC encoder from practical application. Hence, optimizing interprediction is a main challenge for fulfilling coding time saving.

2.1 AMP decision skipping algorithm

Through analyzing comparative results in Section 1, notwithstanding disabling AMP can avoid complicated ME process, most of CUs are further split. Thus, further splitting damages PSNR and bitrates simultaneously. For current CU, more coding time will be saved if the optimal prediction mode is determined in advance. We would like to propose an efficient AMP decision algorithm that can early decide whether AMP is the best partition or not.

According to the simulated results in Table 2, we compare sequence content between AMP on and AMP off. The test sequence "RaceHorses" is taken as an example. We can observe from Fig. 3 that when AMP is disabled, just a small part of CUs carry out SKIP mode (in blue rectangle). While some CUs are partitioned by AMP when AMP is enabled, but their corresponding CUs were split into smaller sub-CUs (in red rectangle) when AMP is disabled. This is the reason that coding time reduces not sufficiently while the loss of coding performance is not so much when AMP is disabled.

As mentioned by [32–34], it is just an MV that distinguishes the current CU and its co-located CU in previous coded frame. Accordingly, the relationship between them can be modeled as $I(p + mv, t - 1) = I(p, t)$, where $I(p, t)$ denotes the value of pixel in p position at t time and mv represents motion vector. Furthermore, the high correlation between CU and its co-located CU has been verified in [33, 34]. In accordance with this high correlation, we conduct extensive experiments on ten sequences in Table 1 to discover the relationship between AMP and CU position. As illustrated in Fig. 4, current CU is termed as CU_{cur}, which should have been encoded as AMP with AMP off. Its corresponding CU is labeled as CU_{cor}, which is decided as AMP. In addition, the co-located CU of CU_{cor} is labeled as CU_{col}. The

Table 2 Comparison between HM 14.0 (AMP off) and the original HM 14.0 (AMP on)

Test sequence	RA			LDP		
	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	BD-PSNR (dB)	BD-BR (%)	ΔT (%)
Traffic	− 0.03	0.78	− 10.45	N/A	N/A	N/A
PeopleOnStreet	− 0.04	0.67	− 9.57	N/A	N/A	N/A
BasketballDrive	− 0.06	1.16	− 16.31	− 0.11	1.21	− 18.12
Cactus	− 0.05	0.98	− 11.06	− 0.06	0.87	− 14.97
BQMall	− 0.05	1.22	− 13.84	− 0.07	0.93	− 12.36
BasketballDrill	− 0.06	1.17	− 15.63	− 0.09	1.08	− 17.21
BQSquare	− 0.02	0.54	− 7.82	− 0.05	0.86	− 15.39
RaceHorses	− 0.06	1.24	− 17.94	− 0.03	0.77	− 13.58
FourPeople	N/A	N/A	N/A	− 0.04	0.82	− 11.32
Johnny	N/A	N/A	N/A	− 0.02	1.13	− 9.42
Average	− 0.05	0.97	− 12.83	− 0.06	0.96	− 14.05

N/A not applicable

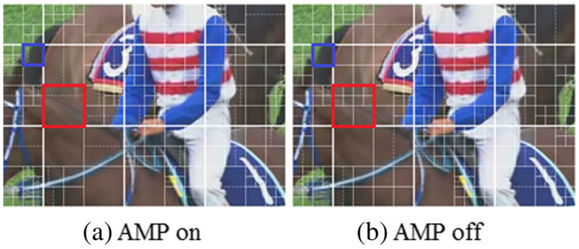

(a) AMP on (b) AMP off

Fig. 3 CU and PU partition of the second frame in "RaceHorses" under RA case with QP of 32. **a** AMP on. **b** AMP off

experiments are to calculate the probabilities of mode distribution of CU_{col} and CU_{cor}.

Table 3 shows the posterior probabilities of CU_{cur} mode selection and CU_{col} mode selection, where M_{cur} and M_{col} represent the optimal mode of CU_{cur} and CU_{col}, respectively. It is obvious from Table 3 that if CU_{cur} are selected as SKIP mode with AMP disabled, over 96% of CU_{col} in previous coded frame is decided as SKIP mode when AMP is enabled. In particular, when QP selects 37, the probability of SKIP mode for CU_{col} reaches up to 98.7%. It demonstrates that if M_{cur} chooses SKIP mode, M_{col} will also be SKIP mode likely. However, if CU_{cur} is further split when AMP is off, the modes on CU_{col} distribute evenly (20.4% of SKIP mode, 18.8% of MERGE mode, 53.1% of three partition modes and 7.7% of intramode). Because of the high relevance between two adjacent frames, AMP decision on current CU can be avoided adaptively based on whether M_{col} is SKIP mode or not. In other words, AMP decision on one CU should be skipped immediately without any RD cost calculation process if the co-located CU in previous coded frame of this CU was encoded as SKIP mode. Otherwise, current CU will be predicted as AMP or still as SMP. This approach achieves some AMP decision skipping with sacrificing coding quality, while this loss is quite negligible and the computational load reduction is very significant.

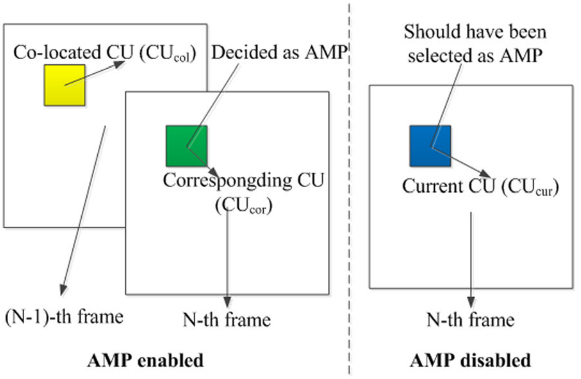


Fig. 4 Corresponding positions of each type CU with AMP enabled and disabled

Table 3 Posterior probabilities of CU_{cur} mode selection and CU_{col} mode selection

M_{cur}	M_{col}	Probability (P(M_{col}|M_{cur}), %)				Average
		QP = 22	QP = 27	QP = 32	QP = 37	
SKIP	SKIP	94.2	95.5	96.3	98.7	96.2
	MERGE	0.8	0.6	0.5	0.1	0.5
	SP	1.1	1.0	0.8	0.4	0.9
	SMP	1.7	1.4	1.2	0.5	1.2
	AMP	1.6	1.3	0.9	0.3	1.0
	Intra	0.6	0.2	0.3	0.0	0.3
Further split	SKIP	18.5	20.1	21.3	21.8	20.4
	MERGE	18.3	19.7	18.1	18.9	18.8
	SP	19.2	20.9	22.1	20.0	20.6
	SMP	19.3	18.4	17.9	16.5	18.0
	AMP	13.4	14.8	14.2	15.7	14.5
	Intra	11.3	6.1	6.4	7.1	7.7

2.2 Search range adjustment algorithm

ME is the most computationally expensive procedure in HEVC encoder. The search range of the matched sample restrains current block to obtain the best MV within a limited area in the reference frame. Larger the search range is, higher the computational burden costs. Contrarily, a very small motion search range contains a few candidate points for matching so that the best MV can be checked out within a short time. An appropriate search range is capable of saving considerable search time and at the same time keeping a good RD performance for HEVC encoder. However, due to small size search range has poor candidates, a low precision and even an extreme error will be generated. In addition, a large motion search range is indispensable for reliable prediction. Therefore, the motion search range can be adjusted properly without loss of HEVC coding efficiency.

In this section, the coding information (including search range and CU size) from the spatial adjacent blocks will be exploited to discover the characteristics among spatial blocks in the same frame and to determine search range for current CU. Extensive experiments were conducted on six typical text sequences from Table 1, and simulated conditions are same as configuration in Section 1.

Six selected test sequences in Table 4 can be encoded with both RA and LDP cases according to CTC. Sequences "BasketballDrive," "BasketballDrill," and "RaceHorses" have many fast moving objects, while sequences "Cactus," "BQMall," and "BQSquare" are with relatively low movement activity, and "BQSquare" even has large static background. For better understanding

Table 4 Distribution of CU sizes for six typical test sequences

Sequences	QP	CU size distribution (area normalization)							
		RA				LDP			
		64 × 64	32 × 32	16 × 16	8 × 8	64 × 64	32 × 32	16 × 16	8 × 8
BasketballDrive	22	0.03	0.08	0.23	0.66	0.03	0.10	0.22	0.64
	27	0.04	0.10	0.26	0.61	0.04	0.11	0.24	0.61
	32	0.05	0.09	0.28	0.58	0.05	0.12	0.24	0.59
	37	0.06	0.11	0.28	0.55	0.05	0.13	0.21	0.61
BasketballDrill	22	0.03	0.08	0.20	0.69	0.03	0.09	0.25	0.63
	27	0.04	0.10	0.24	0.63	0.04	0.10	0.26	0.61
	32	0.04	0.10	0.25	0.61	0.05	0.11	0.25	0.59
	37	0.04	0.10	0.23	0.63	0.05	0.11	0.25	0.59
RaceHorses	22	0.02	0.07	0.29	0.62	0.02	0.13	0.27	0.58
	27	0.03	0.08	0.30	0.59	0.04	0.14	0.31	0.52
	32	0.03	0.08	0.34	0.56	0.03	0.16	0.32	0.49
	37	0.04	0.09	0.31	0.56	0.03	0.16	0.31	0.51
Average		0.04	0.09	0.27	0.61	0.04	0.12	0.26	0.58
Cactus	22	0.32	0.52	0.13	0.04	0.35	0.47	0.11	0.07
	27	0.33	0.48	0.14	0.04	0.37	0.48	0.11	0.05
	32	0.33	0.50	0.13	0.04	0.37	0.48	0.11	0.04
	37	0.35	0.48	0.14	0.03	0.38	0.47	0.12	0.03
BQMall	22	0.34	0.51	0.11	0.04	0.40	0.45	0.11	0.04
	27	0.37	0.50	0.10	0.03	0.41	0.44	0.11	0.05
	32	0.36	0.52	0.09	0.04	0.42	0.43	0.11	0.04
	37	0.38	0.48	0.09	0.04	0.41	0.45	0.10	0.04
BQSquare	22	0.48	0.40	0.08	0.03	0.50	0.40	0.07	0.03
	27	0.44	0.44	0.09	0.03	0.52	0.39	0.06	0.04
	32	0.45	0.46	0.07	0.01	0.48	0.43	0.06	0.03
	37	0.46	0.42	0.10	0.01	0.53	0.38	0.06	0.03
Average		0.38	0.48	0.11	0.03	0.43	0.44	0.09	0.04

for readers, based on the average portions of each CU size, we normalize the distribution data using area normalization method. The area normalization method can be used to measure the fineness of specimen in chemistry, while we adopt this method to further highlight the CU size distribution in this paper. For the sequences with complex motion objects ("BasketballDrive," "BasketballDrill," and "RaceHorses"), their small size CU (16 × 16 and 8 × 8) is close to 0.90 under RA configuration, as well as 0.84 under LDP configuration as shown in Table 4. In particular, as a test sequence with large global motion area, sequence "RaceHorses" has a huge amount of movement information, so the small size CU in this sequence also accounts for nearly 0.90 at RA condition. It is because that motion in these test sequences is quite rich and objects in them move dramatically, it needs small size CU to record abundant movement information for fulfilling an accurate MV coding. Nevertheless, in the other three test sequences with comparative simple motion, the overwhelming majority of CU is big size (64 × 64 and 32 × 32), where it is from 0.83 ("Cactus" in RA case) to 0.90 ("BQSquare" in LDP case) on average at both two conditions. Particularly "BQSquare" has large motionless region, so CUs in this test sequence are almost totally chosen as big size. Another observation from Table 4 is that as QP value is rising, the share of big size CU is growing, since growth of QP smoothens the video texture so that the motion degree is slow. The region in current CU with big size is covered by homogeneous texture or motion content, while the small size CUs carry a large number of texture and movement information. As a result, we can initially determine motion degree upon the current CU size.

Based on aforementioned observations, we can find that motion degree is relative closely to CU size. Moreover, the first three sequences contain complex motion, so the CU size in these four sequences always tends to be small. While the motion in the last three sequences is relatively slow, therefore, the big CU accounts for the vast majority. In other words, fast/slow motion leads to small/big size CU and requires big/small motion search range. Consequently, according to the motion information of CU size, various CU sizes are divided into four categories and the weight value (ω) can be setup on different CU sizes, as shown in Table 5.

MVs of the current CU are always correlated with the MVs of its spatially neighboring blocks [33]. It is possible that the current block and its spatially adjacent blocks have the same moving object with similar movement information and the motion of the object is unlikely to alter abruptly over time. Accordingly, the relationship among movement information of the spatially neighboring blocks can be utilized to adjust search range for current CU. Furthermore, a plenty of experiments have been conducted on various test sequences, and it is observed that most objects move along horizontal or vertical direction. Hence, we tabulate Table 6 to assign the relationship factor (β_i) between current CU and its adjacent spatial blocks. The spatial correlation is one of the major parameters for measuring motion degree of current CU [35]. Additionally, another crucial parameter to judge the degree is current CU size, where different sizes are assigned different weight values as tabulated in Table 5. Therefore, we can determine the motion degree of current CU in accordance with the weight value (ω) and the relationship factors (β_i) as follows:

$$MD = \omega \sum_{i=0}^{3} \omega_i \cdot \beta_i \cdot f_i \tag{2}$$

where MD represents motion degree of current CU, and weight value ω_i is signified by search ranges of neighboring block i according to results in Table 5, as well as the flag f_i is set as 1 if neighboring block i is available; otherwise, it is assigned as 0.

In general, the larger the parameter MD is, the stronger the uncertainty of motion is and the bigger the search range is. According to this regulation, the search range of current CU is able to be determined through Table 7. T_1, T_2, and T_3 in Table 7 are thresholds to classify current CU into each types. Inspired by [35] and through a great deal of experimental results, we achieve the accuracies of the proposed algorithm using various thresholds, which are shown in Table 8. The accuracies here are defined as the matching rate between the MVs by the proposed algorithm and the MVs by the original HEVC encoder. T_1, T_2, and T_3 are assigned to 0.65, 10.50, and 13.00, respectively. This configuration is capable of adjusting search range for current CU, which avoids extremely high computational load from full search range and achieves a consistent coding performance.

Table 5 Weight values for each CU size

CU size	Weight value (ω)
64 × 64	0.1
32 × 32	0.2
16 × 16	0.3
8 × 8	0.4

Table 6 Relationship factor (β_i) and position of neighboring block

i	Relationship factor (β_i)	Position of neighboring block
0	4	Left block
1	1	Top-left block
2	4	Top block
3	1	Top-right block

2.3 Overall algorithms

On the basis of the aforementioned analysis, our proposed efficient approach for HEVC combines fast PU partition decision algorithm and fast ME process algorithm. The flow chart of our proposed overall algorithms is illustrated as Fig. 5.

We further describe the steps in detail as follows:

Step 1: Start interprediction for a CU.

Step 2: Determine whether current CU is partitioned to SMP (PART_2N × N and PART_N × 2 N) or not. If current CU is divided into SMP, go to step 3; otherwise, go to step 4.

Step 3: Perform AMP decision skipping. Derive the co-located CU in previous coded frame of current CU; if it was encoded as SKIP mode, current CU should also be decided as SKIP mode and go to step 6; otherwise, current CU should further be decided as AMP or still as SMP and go to step 4.

Step 4: Perform fast search range adjustment.

Step 4.1: Set the weight value (ω) upon each CU sizes and assign each search ranges corresponding to different ω as shown in Table 5.

Step 4.2: Compute motion degree of current CU based on Eq. (2) using several parameters, including relationship factor from Table 6.

Table 7 Search range adjustment using *MD*

Search range	Condition
[− 8, 8]	$MD < T_1$
[− 16, 16]	$T_1 \le MD < T_2$
[− 32, 32]	$T_2 \le MD < T_3$
[− 64, 64]	$T_3 \le MD$

Table 8 Comparison between ADS and the original HM 14.0

Test sequence	RA			LDP		
	BD-PSNR (dB)	BD-BR (%)	ΔT (%)	BD-PSNR (dB)	BD-BR (%)	ΔT (%)
Traffic	− 0.03	0.47	− 8.26	N/A	N/A	N/A
PeopleOnStreet	− 0.03	0.51	− 8.38	N/A	N/A	N/A
BasketballDrive	− 0.03	0.73	− 13.84	− 0.08	0.91	− 14.33
Cactus	− 0.05	0.71	− 9.93	− 0.05	0.68	− 12.18
BQMall	− 0.04	0.82	− 10.41	− 0.06	0.72	− 10.07
BasketballDrill	− 0.02	0.87	− 12.55	− 0.06	0.85	− 13.46
BQSquare	− 0.02	0.42	− 5.97	− 0.05	0.71	− 11.25
RaceHorses	− 0.04	0.69	− 14.02	− 0.02	0.57	− 12.53
FourPeople	N/A	N/A	N/A	− 0.03	0.63	− 9.76
Johnny	N/A	N/A	N/A	− 0.02	0.89	− 8.87
Average	− 0.03	0.65	− 10.42	− 0.05	0.75	− 11.56

N/A not applicable

Step 4.3: According to values of T_1, T_2, and T_3, decide which search range for the current CU should be selected to execute ME.

Step 5: Perform ME with modified search range for current CU.

Step 6: Go to step 1 and address the next CU.

3 Results and discussion

This section analyzes the performance in terms of coding time, bitrates, and PSNR. In the following, the two sub-methods are described as (i) AMP decision skipping algorithm (labeled “ADS”) and (ii) search range adjustment algorithm 2 (labeled “SRA”).

For both algorithms, the HEVC test model (HM) reference software HM14.0 has been utilized. For these tests, ten sequences are released by JCT-VC group and the detailed information of the test sequences is provided in Table 1 in Section 1. It is noteworthy that all the ten test sequences in Table 1 are selected from the CTC and they have all kinds of resolutions (from 416 × 240 to 2560 × 1600) and various motion characteristics (low and high motion degree). All the experiments are

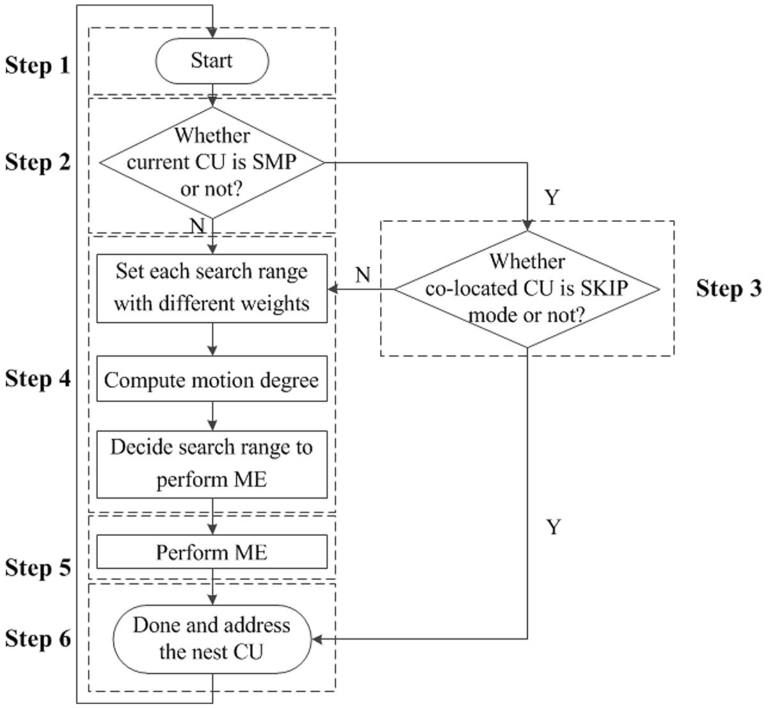

Step 1
Start
Step 2
Whether current CU is SMP or not?
N
Y
Step 3
Whether co-located CU is SKIP mode or not?
N
Set each search range with different weights
Step 4
Compute motion degree
Decide search range to perform ME
Y
Step 5
Perform ME
Step 6
Done and address the nest CU

Fig. 5 Flow chart of our proposed overall algorithms

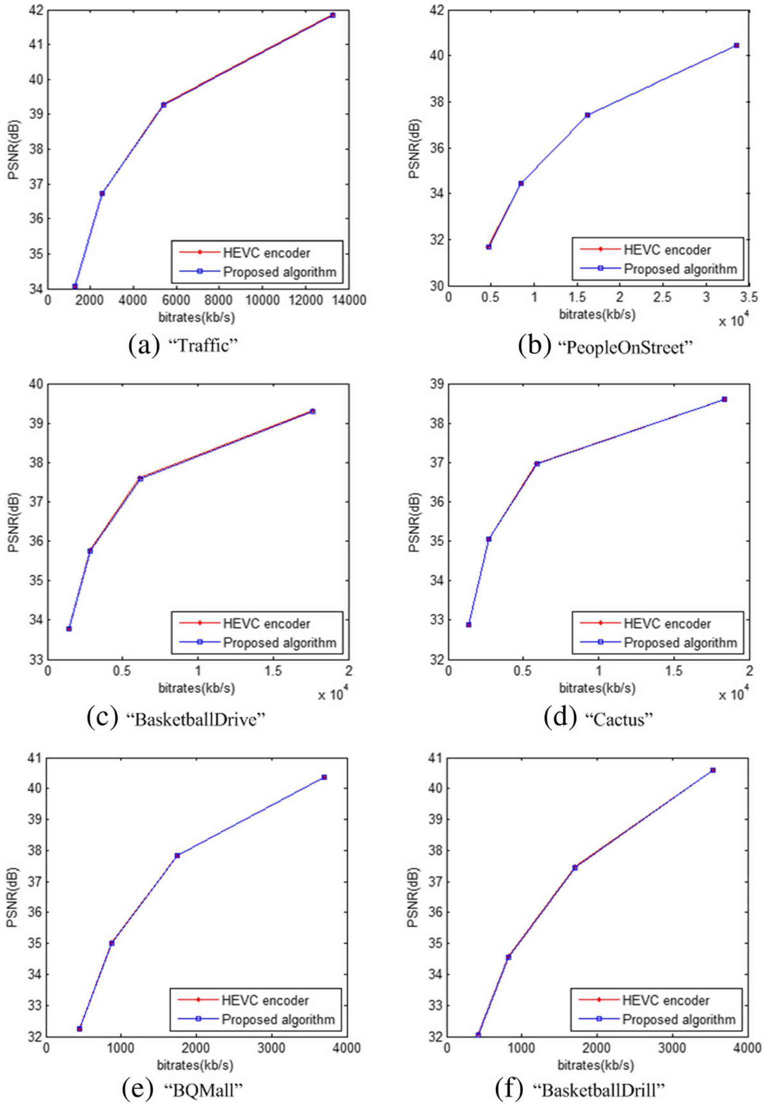


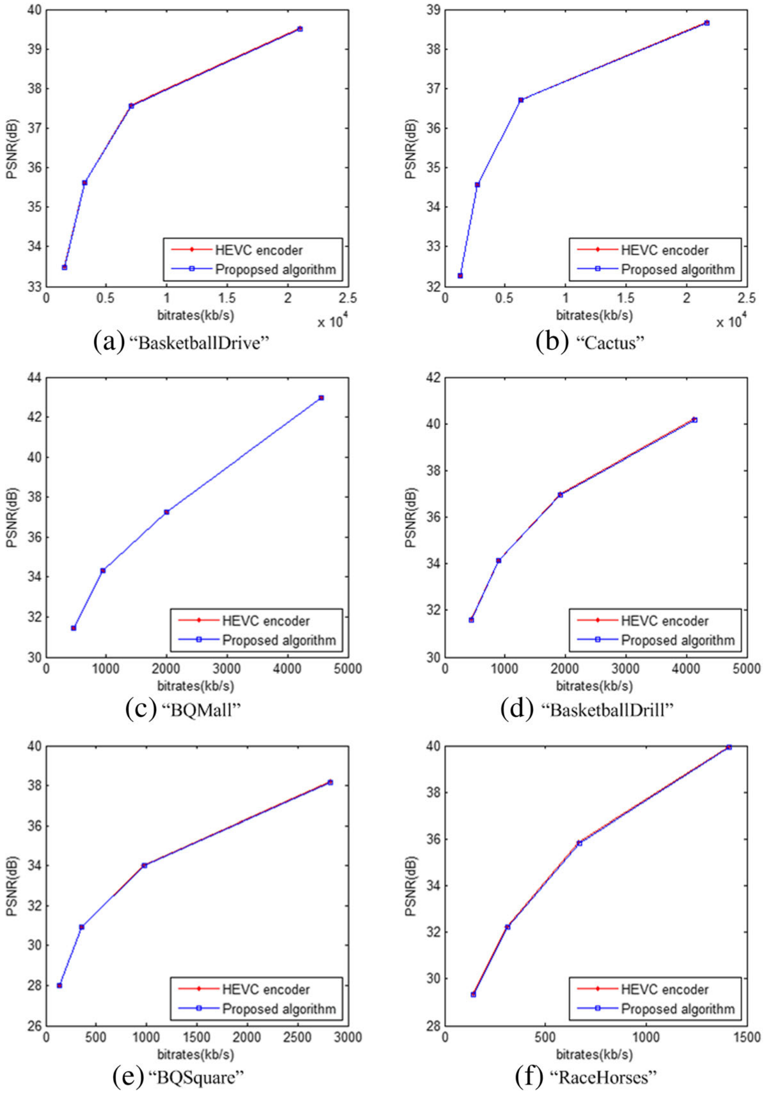
Fig. 6 RD curves with the QP of 22, 27, 32 and 37 under RA configuration. **a** "Traffic". **b** "PeopleOnStreet". **c** "BasketballDrive". **d** "Cactus". **e** "BQMall". **f** "BasketballDrill"

simulated on two Intel Xeon E5-2640 v2 2.0GHz, 32GB DDR3 random access memory, and compiled on Microsoft Visual C++ 2010 under RA and LDP cases. In addition, operating system is 64-bit Microsoft Windows 7 SP1. The encoder is configured as follows: both maximum CU width and height are 64 pixels, and maximum partition depth is 4, resulting in minimum CU size of 8 × 8 pixels; period of I-Frame is 32, and GOP (group of picture) size is 8; full search range of ME is set with 64; QP is respectively selected with 22, 27, 32, and 37; RDO, SAO (sample adaptive offset), and AMP are enabled; and CABAC (context-based adaptive binary arithmetic coding) is used as the entropy coder.

In this section, intuitive comparison is shown in Figs. 6, 7, 8, and 9. Meanwhile, the results of comparative experiments are tabulated in Tables 8, 9, and 10 which show the results between our approach and the original HEVC encoder. Figures 10, 11, and 12 provide the results compared with the state-of-the-art fast intercoding methods in detail. Note that the same as Section 1, changes of PSNR, bitrates, and coding time are labeled by "BD-PSNR (dB)," "BD-BR (%)," and "ΔT (%)," respectively.

3.1 Evaluation of individual algorithm ADS compared with HM 14.0

To assess the coding performance of our proposed ADS algorithm compared with the original HEVC encoder, we have conducted a large number of experiments, and the experimental results in Table 8 illustrate the relative encoding performance of the proposed ADS algorithm



(a) “BasketballDrive”

(b) “Cactus”

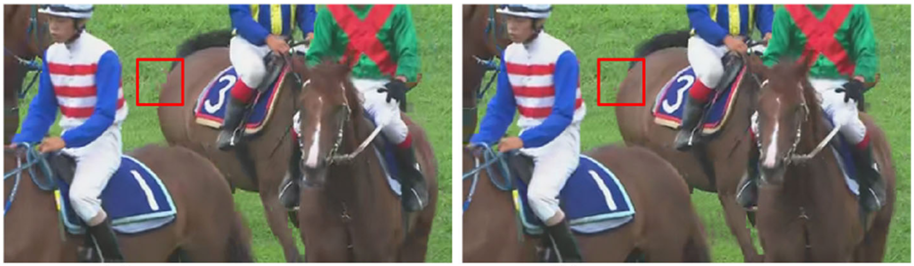
(c) “BQMall”

(d) “BasketballDrill”

(e) “BQSquare”

(f) “RaceHorses”

Fig. 7 RD curves with the QP of 22, 27, 32 and 37 under LDP configuration. **a** “BasketballDrive”. **b** “Cactus”. **c** “BQMall”. **d** “BasketballDrill”. **e** “BQSquare”. **f** “RaceHorses”



(a) encoded by the original HM 14.0

(b) encoded by the proposed algorithm

Fig. 8 Comparison of the subjective quality of the 22nd frame in “RaceHorses”. **a** Encoded by the original HM 14.0. **b** Encoded by the proposed algorithm

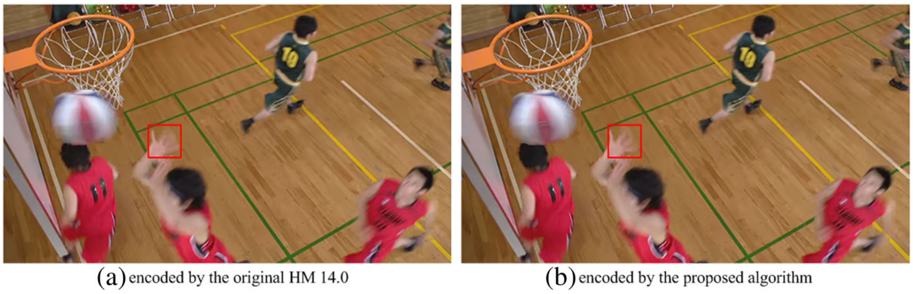

(a) encoded by the original HM 14.0
(b) encoded by the proposed algorithm

Fig. 9 Comparison of the subjective quality of the 32nd frame in test sequence "BasketballDrill". **a** Encoded by the original HM 14.0. **b** Encoded by the proposed algorithm

versus the original HM 14.0. From Table 8, we can note that sub-algorithm ADS can obtain coding complexity reduction with almost the same encoding performance for all ten test sequences. More specifically, it reduces 10.42 and 11.56% coding time averagely for them under RA and LDP cases, respectively. The loss of PSNR for ADS is from – 0.08 dB ("BasketballDrive" under LDP configuration) to – 0.02 dB, and bitrate increase is from 0.42% ("BQSquare" under RA configuration) to 0.91% ("BasketballDrive" under LDP configuration). Since ADS simplifies AMP decision for PU mode decision, for the test sequences with high movement region, coding time saving is from the maximum of 14.33% ("BasketballDrive" at LDP condition) to the minimum of 12.53% ("RaceHorses" at LDP condition), which is evident. Although some of these ten test sequences have much slow motion areas (including "Traffic," "Cactus," "BQMall," "BQSquare,""FourPeople," and "Johnny"), the coding time reduction still achieves approximately 8.67% under RA case and 10.43% under LDP case. Because of high definition and high motion activity, RD performance for test sequence "BasketballDrive" is less than satisfactory, but coding time saving reaches the expected goal with the maximum of 14.33% at LDP condition. These results imply that the proposed ADS algorithm can accelerate PU partition decision in HEVC encoder through skipping some AMP decision.

3.2 Evaluation of individual algorithm SRA compared with HM 14.0

Table 9 demonstrates the experimental results of our SRA algorithm compared with the original HM 14.0. It can be seen from Table 9 that over 16% of time saving is obtained by SRA algorithm at both RA and LDP conditions. Meanwhile, decrease of PSNR and increase of bitrates are within acceptable range (– 0.02 dB of PSNR decrease and 0.37% of bitrates increase at RA condition as well as – 0.01 dB of PSNR decrease and 0.28% of bitrates increase at LDP condition on average). In particular, for test sequences "BQSquare" with RA configuration and "Johnny" with LDP configuration, coding performance are totally identical to that of the original HEVC encoder in respect of PSNR. Additionally, bitrates rise from just 0.20% ("BQSquare" under LDP case) to 0.40% ("BQMall" under RA case) using SRA method. Because SRA adaptively diminishes the motion search range of ME and discards lots of unnecessary search region, bitrate increase is

Table 9 Comparison between SRA and the original HM 14.0

Test Sequence	RA			LDP		
	BD-PSNR (dB)	BD-BR (%)	Δ*T* (%)	BD-PSNR (dB)	BD-BR (%)	Δ*T* (%)
Traffic	– 0.02	0.39	– 15.52	N/A	N/A	N/A
PeopleOnStreet	– 0.01	0.37	– 17.28	N/A	N/A	N/A
BasketballDrive	– 0.02	0.29	– 19.36	– 0.02	0.38	– 18.70
Cactus	– 0.02	0.32	– 16.40	– 0.01	0.27	– 15.56
BQMall	– 0.03	0.40	– 15.07	– 0.01	0.23	– 14.85
BasketballDrill	– 0.01	0.28	– 18.44	– 0.02	0.36	– 19.44
BQSquare	– 0.00	0.37	– 13.87	– 0.01	0.20	– 13.28
RaceHorses	– 0.02	0.27	– 19.13	– 0.02	0.31	– 17.73
FourPeople	N/A	N/A	N/A	– 0.01	0.26	– 16.51
Johnny	N/A	N/A	N/A	– 0.00	0.25	– 16.18
Average	– 0.02	0.37	– 16.88	– 0.01	0.28	– 16.53

N/A not applicable

Table 10 Comparison between ADS + SRA and the original HM 14.0

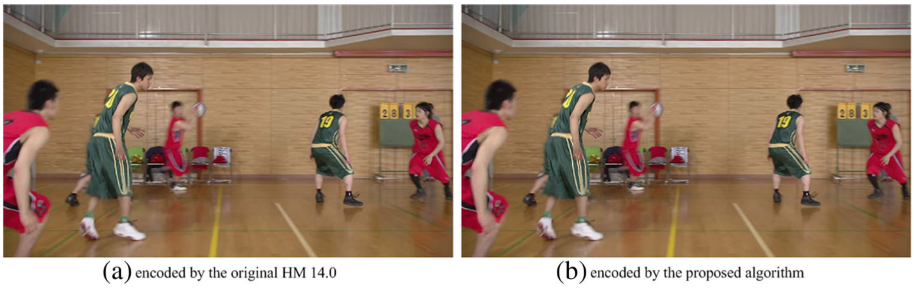
Test Sequence	RA			LDP		
	BD-PSNR (dB)	BD-BR (%)	Δ*T* (%)	BD-PSNR (dB)	BD-BR (%)	Δ*T* (%)
Traffic	– 0.01	0.28	– 29.13	N/A	N/A	N/A
PeopleOnStreet	0.00	0.25	– 27.24	N/A	N/A	N/A
BasketballDrive	– 0.01	0.23	– 30.27	– 0.02	0.20	– 32.59
Cactus	0.00	0.19	– 32.46	0.01	0.17	– 33.41
BQMall	0.00	0.16	– 33.89	0.00	0.14	– 34.82
BasketballDrill	– 0.01	0.20	– 31.69	– 0.01	0.15	– 32.35
BQSquare	0.01	0.13	– 35.58	0.01	0.11	– 35.79
RaceHorses	– 0.01	0.18	– 30.72	0.01	0.16	– 34.36
FourPeople	N/A	N/A	N/A	0.00	0.10	– 35.03
Johnny	N/A	N/A	N/A	0.01	0.09	– 37.21
Average	0.00	0.20	– 31.37	0.00	0.14	– 34.45

inconspicuous with regard to the test sequences with low motion activity areas, such as “BQSquare,” “FourPeople,” and “Johnny.” It indicates that SRA is suitable for the test sequence whose objects move slowly, since SRA method can eliminate much computational complexity on candidate point searching through adjusting search range. Furthermore, coding time reduction is particularly significant on test sequences with complex movement, such as “BasketballDrive,” “BasketballDrill,” and “RaceHorses.” The analysis about the results in Table 9 states that the proposed fast SRA algorithm maintains the nearly same coding efficiency as the original HEVC encoder with considerable reduction for the computational load.

3.3 Proposed overall algorithm assessment

Table 10 illustrates the experimental result comparison between the proposed overall algorithms (ADS plus SRA) and the original HEVC encoder. The results demonstrate that the proposed algorithm can save the encoding time dramatically with 31.37% under RA configuration and 34.45% under LDP configuration on average, where the maximum reduction of 37.21%(“Johnny” at LDP condition) and the minimum reduction of 27.24% (“PeopleOnStreet” at RA condition), respectively. When ADS performs, inter partition mode can be decided in advance. This process has accelerated inter-mode decision. Based on this improvement, ME can be skipped conditionally or performed under proper search range with SRA. Only performing ADS can skip some interpartition modes without any RD cost calculation process, but ME with full search range, if any, consumes much time. Similarly, although SRA can adjust search range for ME, computational complexity is still high during interpartition mode decision without ADS. It is inevitable that collaboration of ADS and SRA outperforms the individual algorithms. Therefore, the coding time reduction by joint algorithm is more than the time saved by ADS and SRA, respectively (11 and 16%).

Test sequence “Johnny” has both large static background regions and large homogeneous areas, so most of CUs can be immediately predicted as SKIP mode without AMP decision using ADS. Furthermore, search range can be diminished as minimum range [– 8, 8] using SRA for some blocks, which accomplishes coding time decrease efficiently. Since the proposed overall algorithms skip complicated AMP decision and adjust motion search range of ME process rather than exhaustive candidates search, the computational burden for the


(a) encoded by the original HM 14.0

(b) encoded by the proposed algorithm

Fig. 10 Comparison of the subjective quality of the 42nd frame in test sequence “BasketballDrive” **a** Encoded by the original HM 14.0. **b** Encoded by the proposed algorithm

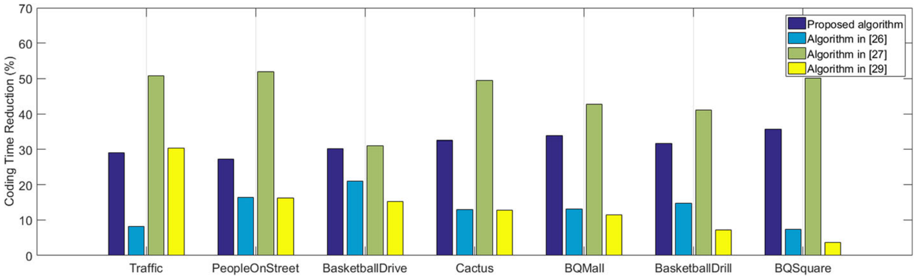


Fig. 11 Coding time reduction comparison of the proposed algorithm and the algorithms in [26, 27, 29] in RA case

sequences with low motion activity reduces significantly. For the sequences with slow movement, including "Traffic," "Cactus," "BQMall," "BQSquare," "FourPeople," and "Johnny," the proposed overall algorithms averagely save coding time 32.77% (in RA case) and 35.25% (in LDP case), and bitrate increase for these sequences are only 0.19% (in RA case) and 0.12% (in LDP case) on average, which is negligible. Moreover, the average PSNR loss for all the test sequences is 0.00 dB under two configurations. These results indicate that the proposed efficient AMP decision and search range adjustment algorithm can accelerate intercoding greatly with the same coding performance as HEVC encoder.

3.4 Intuitive evaluation of our proposed algorithm

In order to illustrate the experimental results intuitively, Figs. 6 and 7 provide more information about the proposed overall algorithm compared with the original HEVC encoder in detail. It can be noticed from Figs. 6 and 7 that the proposed overall algorithm can obtain nearly the same coding performance as the original HEVC encoder in terms of PSNR and bitrates.

3.5 Subjective performance comparison

Figures 8, 9, and 10 show the 22nd frame of "RaceHorses" (416 × 240), the 32nd frame of "BasketballDrill" (832 × 480), and the 42nd frame of "BasketballDrive" (1920 × 1080) under RA case with QP of 32, aiming at comparing subjective performance between our proposed approach and the original HEVC encoder.

It is evident that the region marked by the red rectangle in Fig. 8b is more smoothly than that in Fig. 8a. Additionally, the area labeled by the red rectangle in Fig. 9a is more blurred than that in Fig. 9b. Except for Figs. 8 and 9, we can also find from Fig. 10 that there exists no apparent flaw in Fig. 10b for the proposed algorithm from the perspective of subjective quality. The subjective performance comparison demonstrates that the proposed algorithm and the original HEVC encoder have extremely similar subjective qualities, but the coding efficiency of the proposed algorithm is higher than that of the original HEVC encoder.

3.6 Comparison with the state-of-the-art algorithms [26, 27, 29]

It is illustrated as Figs. 11, 12, and 13 that the coding efficiency comparison between the proposed algorithm and the state-of-the-art algorithms in Ref. [26, 27, 29] under RA configuration and Figs. 11, 12, and 13 reflect the comparisons about coding time reduction, Bjontegaard Delta PSNR, and Bjontegaard Delta Bitrate, respectively. From Fig. 11, it is obvious that other than the algorithm in [27], the coding time is generally saved more by the proposed algorithm than that by algorithms in [26, 29], but the bitrate increase by our method is significantly less than that by the method in [27] (as shown in Fig. 13). Meanwhile, according to the

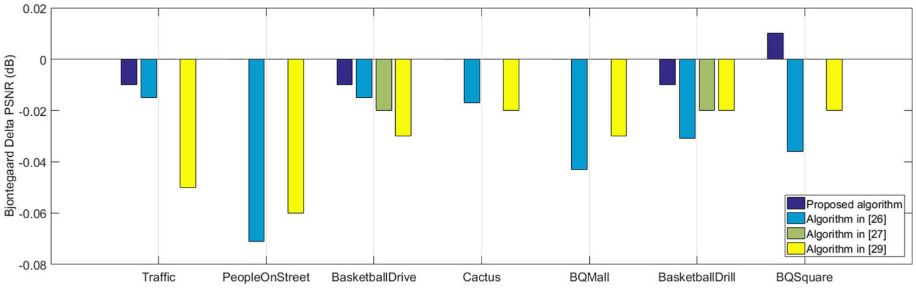


Fig. 12 PSNR change comparison of the proposed algorithm and the algorithms in [26, 27, 29] in RA case (the blank is zero that means no change)

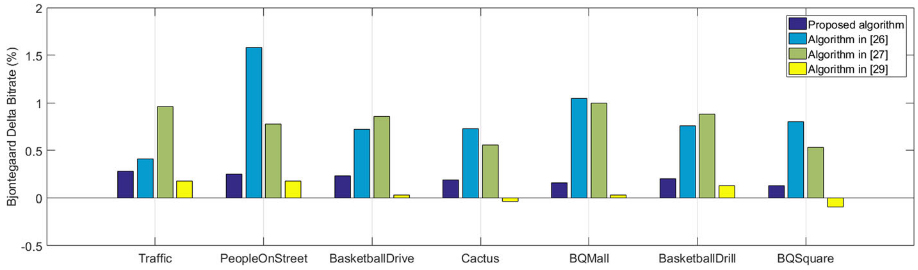


Fig. 13 Bitrate increase comparison of the proposed algorithm and the algorithms in [26, 27, 29] in RA case

data comparisons from Fig. 12, we can see that in respects of PSNR loss achievement of our proposed algorithm is nearly same as that of algorithm in [27]. Note that the proposed algorithm distinctly outperforms the other algorithms in [26, 29]. For bitrate drop, our method is superior to both algorithms in [26, 27] except for algorithm in [29], while our approach achieves a better coding time reduction and PSNR gain than the approach in [29]. To sum up, the proposed algorithm can keep a consistent coding efficiency compared with the state-of-the-art fast HEVC algorithms in [26, 27, 29].

4 Conclusion

In this paper, our proposed algorithm performs the outstanding coding efficiency for interprediction and is demonstrated by aforementioned experimental results. Because the proposed method achieves redundant PU mode decision skipping and adaptive search range of ME adjustment, it can reduce abundant coding time extremely and maintain almost the same coding performance as the original HEVC encoder. Moreover, comparison experiments with the state-of-the-art algorithms in Ref. [26, 27, 29] show that our algorithm is more close to the original HEVC encoder than other novel methods in terms of RD performance. In conclusion, the proposed algorithm can improve coding efficiency significantly and implement HEVC encoder for real-time application.

Acknowledgements

The authors would like to thank the editors and anonymous reviewers for their valuable comments.

Funding

This work was supported in part by the National Natural Science Foundation of China, under Grants U1301257, 61571285, and 61771432.

Availability of data and materials

Data will not be shared; reason for not sharing the data and materials is that the work submitted for review is not completed. The research is still ongoing, and those data and materials are still required by the author and co-authors for further investigations.

Authors' contributions

PA designed and conceived the research. XP performed the simulated experiments and QZ analyzed the exprimental results. XP wrote the manuscript. PA and QZ edited the manuscript. All authors read and approved the final manuscript.

Authors' information

Xinpeng Huang received the M.D. degree in information engineering from Zhengzhou University of Light Industry, Zhengzhou, China, in 2013. He is currently pursuing the Ph.D. degree in communication and information systems from Shanghai University, Shanghai, China. His current research interests include video coding, light field image compression, and extensions of the High Efficiency Video Coding.

Ping An received her B.S. and M.S. degrees from Hefei University of Technology, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree in communication and information systems from Shanghai University, Shanghai, China, in 2002. She is currently a professor in School of Communication and Information Engineering, Shanghai University. Her research interests include stereoscopic and three-dimensional vision analysis and image and video processing, coding, and application.

Qiuwen Zhang received his Ph.D. degree in communication and information systems from Shanghai University, Shanghai, China, in 2012. Since 2012, he has been with the faculty of the College of Computer and Communication Engineering, Zhengzhou University of Light Industry, where he is currently an Associate Professor. He has published over 30 technical papers in the field of video coding and image processing. His major research interests include 3D signal processing, 3D High Efficiency Video Coding (3D-HEVC), video codec optimization, and multimedia communication.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China. [2]Key Laboratory of Ministry of Education for Advanced Display and System Application, Shanghai University, Shanghai 200072, China. [3]College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China.

Received: 29 March 2017 Accepted: 7 November 2017
Published online: 21 November 2017

References

1. B Bross, W-JH, GJ Sullivan, J-R Ohm, T Wiegand. High Efficiency Video Coding (HEVC) text specification draft 8, in document JCTVC-J1003, Stockholm, Switzerland, 2012. http://phenix.intevry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip.
2. GJ Sullivan, J-R Ohm, W-J Han, T Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on circuits and systems for video technology **22**(12), 1649–1668 (2012)
3. G Corrêa, P Assunção, L Agostini, LADS Cruz, *Performance and Computational Complexity Assessment of HEVC* (Springer International

Publishing, Cham, Switzerland, 2016). https://link.springer.com/content/pdf/10.1007%2F978-3-319-25778-5.pdf.

4. Rao, K.R.: High efficiency video coding. 2016 Signal Processing: Algorithms, Architectures, Arrangements and Applications (SPA), 11-11 (2016). doi:10.1109/spa.2016.7763576
5. Mora EG, Cagnazzo M, Dufaux F. AVC to HEVC transcoder based on quadtree limitation. Multimedia Tools & Applications **76**(6), 1-25 (2017)
6. G Tech, Y Chen, K Müller, J-R Ohm, A Vetro, Y-K Wang, Overview of the multiview and 3D extensions of High Efficiency Video Coding. IEEE Transactions on Circuits and Systems for Video Technology **26**(1), 35–49 (2016)
7. DA Milovanovic, D Kukolj, ZS Bojkovic, *Recent Advances on 3D Video Coding Technology: HEVC Standardization Framework* (Springer New York, New York, USA, 2017). https://link.springer.com/content/pdf/10.1007%2F978-1-4939-4026-4.pdf.
8. F Bossen, B Bross, K Suhring, D Flynn, HEVC complexity and implementation analysis. IEEE Transactions on Circuits & Systems for Video Technology **22**(12), 1685–1696 (2012)
9. Yu, H., Wang, W., Xu, M.: Context Reduction of Palette Run Type in High Efficiency Video Coding (HEVC) Screen Content Coding (SCC). (2017).
10. P Wen-Hsiao, FG Walls, RA Cohen, X Jizheng, J Ostermann, A MacInnis, L Tao, Overview of screen content video coding: technologies, standards, and beyond. IEEE J. Emerg. Sel. Top. Circuits Syst. (USA) **6**(4), 393–408 (2016). doi: 10.1109/jetcas.2016.2608971
11. W Liu, J Li, YB Cho, A novel architecture for parallel multi-view HEVC decoder on mobile device. EURASIP Journal on Image and Video Processing **2017**(1), 24 (2017). doi:10.1186/s13640-017-0174-5
12. Y Dai, D Liu, F Wu, *A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding* (Springer International Publishing, Cham, Switzerland, 2017). https://link.springer.com/content/pdf/10.1007%2F978-3-319-51811-4.pdf.
13. A BenHajyoussef, T Ezzedine, A Bouallègue, Gradient-based pre-processing for intra prediction in High Efficiency Video Coding. EURASIP Journal on Image and Video Processing **2017**(1), 9 (2017). doi:10.1186/s13640-016-0159-9
14. Roh HJ, Han SW, Ryu ES. Prediction complexity-based HEVC parallel processing for asymmetric multicores, 1-14, (2017). doi: 10.1007/s11042-017-4413-7.
15. Zupancic, I, Blasi SG, Peixoto E, Izquierdo E. HEVC encoder optimisations using adaptive coding unit visiting order. In: IEEE International Conference on Image Processing 2016, pp. 794-798. http://ieeexplore.ieee.org/document/7532466/.
16. Qing A, Zhou W, Wei H, Zhou X, Zhang G, Yang J. A fast CU partitioning algorithm in HEVC inter prediction for HD/UHD video. In: 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1-5. http://ieeexplore.ieee.org/document/7820718/.
17. Z Liu, X Yu, Y Gao, S Chen, X Ji, D Wang, CU partition mode decision for HEVC hardwired intra encoder using convolution neural network. IEEE Trans. Image Process. **25**(11), 5088–5103 (2016). doi:10.1109/TIP.2016.2601264
18. G Correa, PA Assuncao, LV Agostini, LA da Silva Cruz, Fast HEVC encoding decisions using data mining. IEEE Transactions on Circuits and Systems for Video Technology **25**(4), 660–673 (2015)
19. Zeng H, Xiang W, Cai C, Chen J. Hierarchical inter mode decision for HEVC. In: International Symposium on Intelligent Signal Processing and Communication Systems 2015, pp. 658-662. http://ieeexplore.ieee.org/document/7432853/.
20. W Zhao, T Onoye, T Song, Hierarchical structure-based fast mode decision for H.265/HEVC. IEEE Transactions on Circuits & Systems for Video Technology **25**(10), 1651–1664 (2015)
21. Zupancic, I., Blasi, S.G., Peixoto, E., Izquierdo, E.: Inter-prediction optimizations for video coding using adaptive coding unit visiting order. IEEE Transactions on Multimedia **PP**(99), 1-1 (2016). doi:10.1109/TMM.2016.2579505
22. HS Kim, RH Park, Fast CU partitioning algorithm for HEVC using an online-learning-based Bayesian decision rule. IEEE Transactions on Circuits and Systems for Video Technology **26**(1), 130–138 (2016). doi:10.1109/TCSVT.2015.2444672
23. X Huang, Q Zhang, X Zhao, W Zhang, Y Zhang, Y Gan, Fast inter-prediction mode decision algorithm for HEVC. Signal Image Video Process. **11**(1), 33–40 (2017). https://doi.org/10.1007/s11760-016-0887-4
24. Zhang, N., Fan, X., Zhao, D., Gao, W.: Merge mode for deformable block motion information derivation. IEEE Transactions on Circuits and Systems for Video Technology **PP**(99), 1-1 (2016). doi:10.1109/TCSVT.2016.2589818
25. Nguyen T, Nguyen P, Nguyen P, Dinh C. A novel search pattern for Motion Estimation in High Efficiency Video Coding. In: 2016 International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6. http://ieeexplore.ieee.org/document/7479971/.
26. Z Pan, J Lei, Y Zhang, X Sun, Fast motion estimation based on content property for low-complexity H.265/HEVC encoder. IEEE Trans. Broadcast. **62**, 1–10 (2016)
27. A Medhat, A Shalaby, MS Sayed, M Elsabrouty, Adaptive low-complexity motion estimation algorithm for high efficiency video coding encoder. IET Image Process. **10**(6), 438–447 (2016)
28. Wu J, Guo B, Hou J, Yan Y, Jiang J. Fast CU encoding schemes based on merge mode and motion estimation for HEVC inter prediction. Ksii Transactions on Internet & Information Systems, **10**(3),1195-1211 (2016)
29. H Kibeya, F Belghith, MAB Ayed, N Masmoudi, Fast coding unit selection and motion estimation algorithm based on early detection of zero block quantified transform coefficients for high-efficiency video coding standard. IET Image Process. **10**(5), 371–380 (2016)
30. Bossen F. Common test conditions and software reference configurations, in document JCTVC-F900 Geneva, Switzerland, March, 2011. http://phenix.intevry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F900-v1.zip.
31. G Bjøntegaard. Calculation of average PSNR differences between RD-curves, ITU-T Q.6/SG16 VCEG 13th Meeting, Document VCEG-M33, Austin, USA, 2001
32. J Xiong, H Li, F Meng, S Zhu, Q Wu, B Zeng, MRF-based fast HEVC inter CU decision with the variance of absolute differences. IEEE Transactions on Multimedia **16**(8), 2141–2153 (2014). doi:10.1109/TMM.2014.2356795
33. L Shen, Z Liu, Z Zhang, X Shi, Fast inter mode decision using spatial property of motion field. IEEE Transactions on Multimedia **10**(6), 1208–1214 (2008). doi:10.1109/TMM.2008.2001358
34. L Shen, Z Zhang, Z Liu, Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations. IEEE Transactions on Circuits and Systems for Video Technology **24**(10), 1709–1722 (2014). doi:10.1109/TCSVT.2014.2313892
35. Q Zhang, M Chen, X Huang, N Li, Y Gan, Low-complexity depth map compression in HEVC-based 3D video coding. EURASIP Journal on Image and Video Processing **2015**(1), 2 (2015). doi:10.1186/s13640-015-0058-5