IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Fast prediction mode selection and CU partition for HEVC intra coding

ISSN 1751-9659
Received on 2nd March 2019
Revised 24th July 2019
Accepted on 2nd March 2020
E-First on 8th June 2020
doi: 10.1049/iet-ipr.2019.0259
www.ietdl.org

Bin Fu[1], Qiangqing Zhang[1], Jianling Hu[1] ✉

[1]*School of Electronic and Information Engineering, Soochow University, Jiangsu Suzhou 215006, People's Republic of China*

✉ *E-mail: jlhu@suda.edu.cn*

Abstract: Compared with the predecessor H.264/advanced video coding, high-efficiency video coding (HEVC) is a new video coding standard with nearly double coding efficiency under the same coding quality. However, the computing complexity of HEVC increases sharply. To solve this problem, a fast algorithm for intra prediction mode selection based on mode grouping is proposed by reducing the number of modes entering rough mode decision. Moreover, the dual support vector machine is proposed to efficiently select the coding unit (CU) size, which is based on texture features of CU and sub-CUs including content complexity and direction complexity. By using the new CU size selection algorithm, the encoder can confirm the split for complex CU and terminate the split for simple CU in advance, so as to reduce the computing complexity of CU size selection. The experimental results show that by employing the two fast algorithms in intra coding, it can save 42.80% encoding time, with 0.98% increment in bit rate and 0.018 dB loss of peak-signal-to-noise ratio of luminance, compared with the reference software x265-1.7.

1 Introduction

The increasing popularity of high-definition and ultra-high-definition video has brought better visual experience, meanwhile it also poses a huge challenge to the storage and transmission of video data. So some high-efficiency video codecs, such as VP9, LHE, and high-efficiency video coding (HEVC) [1, 2], have been developed to enhance transmission efficiency and save storage space. Compared with H.264/advanced video coding (AVC), HEVC can reduce the coding rate by about 50% [3, 4]. Intra prediction is an important part of HEVC encoder [5]. HEVC enhances coding performance by introducing 35 intra prediction modes and CU quadtree partition decision [6].

In HEVC's encoding process, the image is first split into several coding tree units (CTUs), which are then split into coding units (CUs) in the form of quadtree [3]. The size of the largest coding unit (LCU) is 64×64, the corresponding split depth is 0 [7]; the size of the smallest coding unit is 8×8, the corresponding depth is 3. It is necessary to traverse the rate distortion (RD) cost calculation of 35 intra prediction modes under each split depth and determine the final prediction mode and CU size through the from-bottom-to-up backtracking process [8]. The computing complexity of the whole process is very high. Fig. 1 illustrates a quadtree partition example of a CTU (LCU) and corresponding CU combination structure. The depth information in Fig. 1 is: CU0-depth 0, CU1-depth 1, CU2-depth 2, CU3-depth 3.

In this study, based on the gradient information of prediction unit (PU) in horizontal, vertical, 45° and 135° directions and the texture features of CUs of different sizes, the fast algorithms for intra prediction mode selection and CU size selection are proposed, respectively, which effectively reduce the computing complexity of intra prediction. The remainder of this paper is organised as follows: in Section 2, related works are presented. Sections 3 and 4 provide an introduction to the prediction mode selection algorithm and CU size selection algorithm. Section 5 gives the experimental results and corresponding analysis. Section 6 concludes this paper.

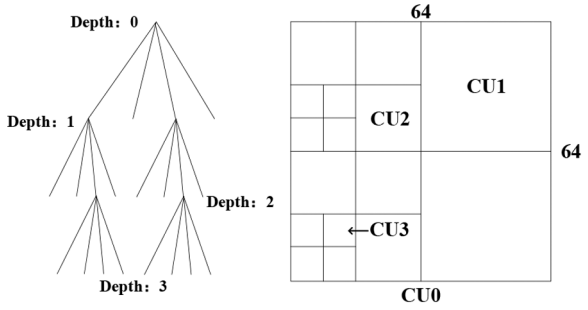

Fig. 1 *Quadtree partitioning and corresponding CU structure*

2 Related works

There are mainly two categories of methods to reduce the computing complexity of intra prediction. One is to optimise the intra prediction mode selection algorithm. Compared with traversing RD costs of 35 intra-prediction modes, rough mode decision (RMD) proposed by Piao *et al.* in 2010 [9] and most probable mode (MPM) proposed by Zhao *et al.* in 2011 [10] effectively reduced the number of modes performing the rate-distortion optimisation (RDO) process and become part of HEVC standard. Many studies are further carried out based on the RMD + MPM + RDO intra prediction mode fast selection algorithm. In 2012, Jiang *et al.* proposed a gradient-based prediction mode selection algorithm. According to the histogram of the gradient vector magnitude, only the modes with the largest amplitudes are selected to perform the RMD process [11]. In 2016, Zhang *et al.* proposed a fast algorithm based on mode grouping. It reduces the number of modes of the RMD process according to the statistical characteristics between the ratio of PU's horizontal gradient to vertical gradient and MPM [8]. In 2017, Maher *et al.* employed block texture as feature to reduce the number of modes of RMD [12]. Apart from reducing the number of modes of the RMD process, some studies focused on directly reducing the number of modes for RDO. In 2013, Chen *et al.* established a list of optimal adjacent modes by analysing the cost of several main direction modes, effectively reducing the number of RDO candidate modes [13]. Zhang *et al.* proposed a similar idea in 2014 based on the Hadamard (HAD) cost, which selectively checked the potential prediction modes instead of traversing all candidate modes [14]. In 2015, Jamali *et al.* proposed a prediction mode selection algorithm based on edge detection and sum of absolute transformed difference cost classification, which increases the prediction accuracy by increasing the weight of the prediction modes near

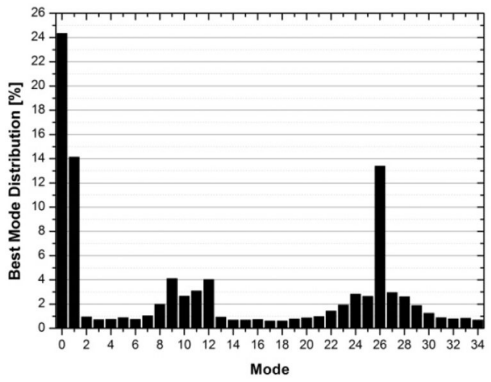

Fig. 2 *Distribution of the best intra prediction mode*

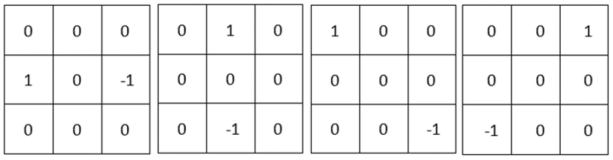

Fig. 3 *Gradient operator*

edge direction [15]. In 2015, Lai and co-worker utilised the best modes of the top and left neighbouring PUs to skip some modes [16]. Reuze *et al.* proposed a method for enhancing intra-mode signalling in 2017 [17]. Zhu *et al.* proposed a fast mode decision algorithm based on texture partition and direction in 2018 [18]. It consists of two sub-algorithms: the CTU depth range prediction and the intra-prediction mode selection.

The other is to optimise the intra CU size selection algorithm. In 2013, Shi *et al.* proposed a CU depth range determination algorithm that uses depth information of spatial and temporal neighbouring CUs [19]. In 2015, Wang *et al.* defined three depth ranges, respectively, for a simple LCU, a complex LCU, and other LCUs. The depth of the current LCU is determined by the maximum depth of adjacent LCUs [20]. In 2017, Sun *et al.* employed the texture complexity of the CU extracted by Haar wavelet transform to perform early CU splitting termination [21]. In 2016, Oztekin *et al.* proposed an early CU determination algorithm based on the strength of pixel variations in picture content [22]. Machine learning has also been widely used in the study of accelerating the CU quadtree partitioning process. In 2013, Cho and Kim [23] proposed a CU fast partitioning and pruning algorithm based on Bayesian classifier using HAD cost and RD cost. In 2015, Lim *et al.* proposed another Bayesian classifier-based CU fast partitioning algorithm, which determines the early split and early termination of CU according to the ratio of HAD costs between current CU to neighbouring CUs or parent CU [24]. In 2017, Hsu *et al.* employed support vector machine (SVM) that uses three parameters as features: variances, low-frequency AC components of discrete cosine transform and spatially neighbouring CU levels for fast CU size decision [25]. In 2018, the authors of [26, 27] employed convolutional neural network to make fast CU depth decision. There are also many studies employing more than one method to optimise intra module, including [16, 28–30]. Tseng and Lai [16] proposed a fast CU decision method and a fast mode selection method. Zhang *et al.* [28] proposed a fast method, which consists of two stages of SVM-based fast CU size decision schemes at four CU decision layers. Jamali and Coulombe [29] proposed a method based on the prediction of the RDO cost of intra modes and employed Prewitt operator to eliminate the non-relevant modes. Shen *et al.* [30] used spatial correlation to skip some depth levels and prediction modes.

3 Intra prediction mode selection fast algorithm

In the 'three-step intra prediction mode fast selection' strategy RMD + MPM + RDO, the RMD and MPM processes reduce the number of modes requiring RDO from 35 to 3–11, which significantly reduced the computing complexity of intra coding. In [8], based on the strategy RMD + MPM + RDO, mode grouping is established according to the gradient information of PU in horizontal and vertical directions, and the number of modes entering the RMD process is reduced. This study also takes the gradient information in the 45° and 135° directions into consideration to further reduce the computing complexity of prediction mode selection.

3.1 Optimal prediction mode distribution

To deeply analyse the intra prediction mode selection process, this study selects four video sequences ParkScene (1080p), Johnny (720p), BQMall (480p) and BasketballPass (240p) in the Joint-Collaborative Team on Video Coding (JCT-VC) standard test video set with different scene and texture features. The distribution of the best intra prediction mode is investigated. The encoder × 265 is configured as full intra coding, quantisation parameter (QP) is set to 32, and for each sequence ten frames are encoded. The statistical result is shown in Fig. 2.

In HEVC, modes 2–17 are called horizontal class mode and modes 18–34 are called vertical class mode. In this study, we call modes 7–13 as horizontal class mode, modes 23–29 as vertical class mode, modes 14–22 as 45° class mode, modes 2–6 and 30–34 as 135° class mode. It can be seen from Fig. 2 that the ratio of planar mode (mode 0) and DC mode (mode 1) is up to 24 and 14%, which means there is a relatively large region of the video images with flat textures after CU quadtree partitioning. In addition to the planar and DC modes, the optimal angle prediction mode focuses on the horizontal mode (mode 10) and the vertical mode (mode 26) and their adjacent modes. This is because the texture information in the vertical or horizontal direction is generally strong in one frame. Meanwhile, PUs also have a ratio of 7 and 8% to select the 45° class mode and the 135° class mode. Therefore, the prediction modes can be grouped according to the characteristics of the optimal prediction mode distribution, to improve the efficiency of intra prediction mode selection.

3.2 Prediction mode grouping

The optimal intra prediction mode has a strong correlation with the direction information of PU. Gradient information can reflect the direction of the PU well. In this study, we use simple gradient operators to calculate the gradient of the luminance component of PU pixel as shown in Fig. 3 (from left to right correspond to horizontal, vertical, 135°and 45° directions, respectively).

In this study, we call the average gradients of prediction unit (PU) in horizontal, vertical, 45° and 135° directions as AGH, AGV, AGMD, and AGDD, respectively. AGH and AGV are the means of the horizontal gradients $|G_x|$ and vertical gradients $|G_y|$ of PU, respectively, shown as (1)–(4), where N represents the size of the current PU, $p(i, j)$ represents the luminance value of a pixel located at (i, j) in PU

$$G_x(i,j) = \begin{cases} p(i,j) - p(i,j+1), & j = 0 \\ p(i,j-1) - p(i,j), & j = N-1 \\ p(i,j-1) - p(i,j+1), & \text{others} \end{cases} \quad (1)$$

$$G_y(i,j) = \begin{cases} p(i,j) - p(i+1,j), & i = 0 \\ p(i-1,j) - p(i,j), & i = N-1 \\ p(i-1,j) - p(i+1,j), & \text{others} \end{cases} \quad (2)$$

$$\text{AGH} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |G_x(i,j)| \quad (3)$$

$$\text{AGV} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |G_y(i,j)| \quad (4)$$

AGMD and AGDD are the means of the 45° gradients $|G_{45°}|$ and 135° gradients $|G_{135°}|$ of the PU, respectively, as shown in (5)–(8)

$$G_{45^\circ}(i,j) = \begin{cases} 0, & j = 0 \text{ or } N-1 \\ 0, & i = 0 \text{ or } N-1 \\ p(i-1,j-1) - p(i+1,j+1), & \text{others} \end{cases} \quad (5)$$

$$G_{135^\circ}(i,j) = \begin{cases} 0, & j = 0 \text{ or } N-1 \\ 0, & i = 0 \text{ or } N-1 \\ p(i-1,j+1) - p(i+1,j-1), & \text{others} \end{cases} \quad (6)$$

$$\text{AGMD} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|G_{45^\circ}(i,j)| \quad (7)$$

$$\text{AGDD} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|G_{135^\circ}(i,j)| \quad (8)$$

The following assumptions are made based on AGH, AGV, AGMD, and AGDD:

(i) If AGV, AGH, AGMD, and AGDD are all small enough, the PU can be considered to be flat and uniform, and the DC or planar mode is very likely to be the best prediction mode.
(ii) If AGV is greater than AGH, the horizontal class mode is more likely to be the best prediction mode and vice versa.
(iii) If AGDD is greater than AGMD, then the 45° class mode is more likely to be the best prediction mode and vice versa.

The probability of selecting 45° class mode φ(AGDD/AGMD) versus AGDD/AGMD and the probability of selection 135° class mode φ(AGMD/AGDD) versus AGMD/AGDD are evaluated using sequences of BasketballPass and BQMall as shown in Figs. 4*a* and *b*, respectively. As shown in Fig. 4, for a PU whose AGDD/AGMD value is >2, it has a probability of more than 80% to be selected as the 45° class mode. On the contrary, for a PU whose AGMD/AGDD value is >2, it has a probability of more than 90% to be selected as the 135° class mode. Therefore, AGDD/AGMD (AGMD/AGDD) can be used to roughly determine the 45° (135°) class mode of PU. Similarly, the judgement of horizontal class and vertical class mode also conforms to such characteristics.

According to AGH, AGV, AGH/AGV, and AGV/AGH, and AGMD, AGDD, AGMD/AGDD, and AGDD/AGMD, this study divides the 35 intra prediction modes into ten groups as shown in Table 1, where group no. indicates the mode group number, modes indicates the modes of intra prediction, and RMD_num indicates the number of modes to enter the RMD.

Each PU selects only one of the mode groups for RMD based on the gradient values in the four directions.

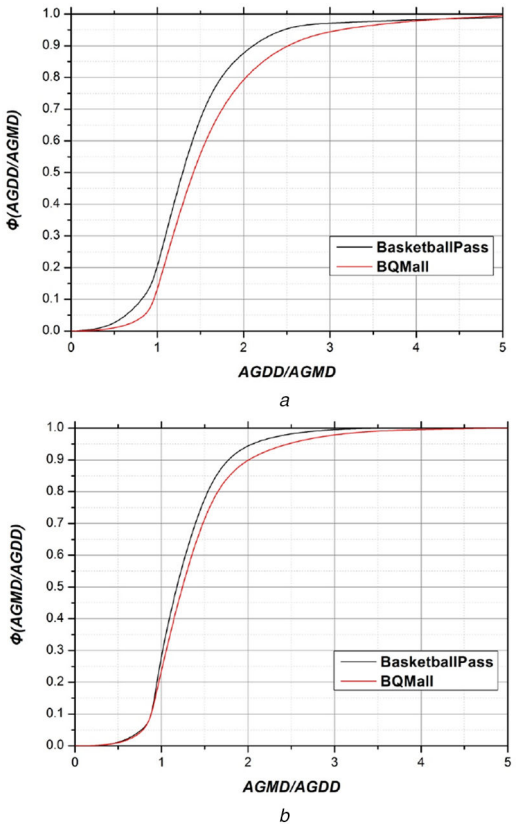

Fig. 4 *Probability distribution function*
(a) AGDD/AGMD, ***(b)*** AGMD/AGDD

Table 1 Mode grouping

group no.	1	2	3	4	5
modes	0, 1	25–27	0, 1, 23–29	9–11	0, 1, 7–13
RMD_num	2	3	9	3	9
group no.	6	7	8	9	10
modes	0, 1, 14–22	0, 1, 2–6, 30–34	0, 1, 18–34	0, 1, 2–18	0–34
RMD_num	11	12	19	19	35

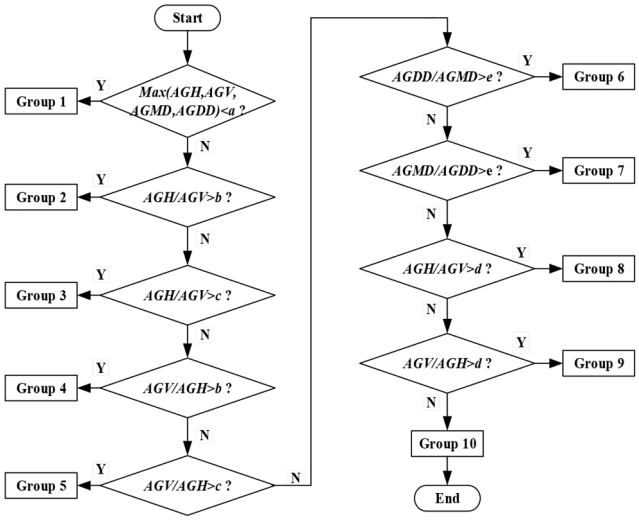

Fig. 5 *Flowchart of the mode group decision procedure*

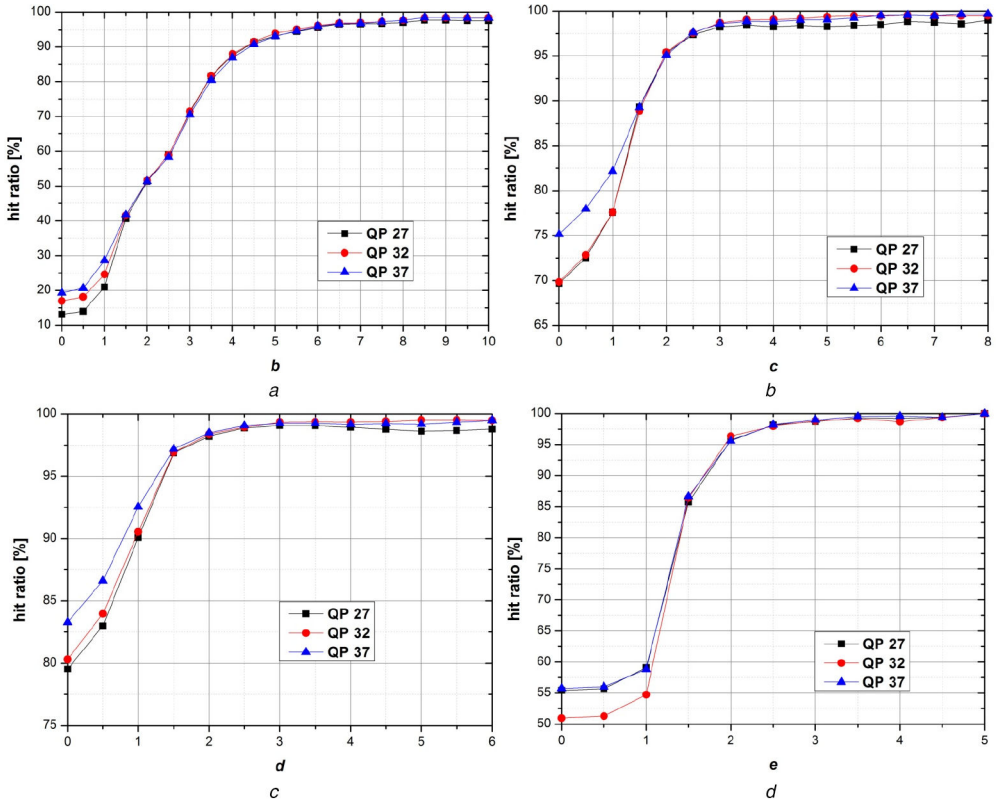

Fig. 6 *Relationship between hit ratio and corresponding Th*
(a) Th *b*, ***(b)*** Th *c*, ***(c)*** Th *d*, ***(d)*** Th *e*

The mode group decision process is depended on five thresholds (Ths), a–e as depicted in Fig. 5, which are determined through experiments.

The determination of Ths *b*–*e* is based on the hit ratios of the best prediction mode. The hit ratio is defined as defining a condition *C* which means that the ratio of gradients is greater than corresponding Th; defining the number of PUs that satisfy condition *C* and whose best prediction modes belong to the corresponding group as *A*; defining the total number of PUs that satisfy *C* as *B*; then the hit ratio equals *A*/*B*. The hit ratios versus Ths *b*–*e* are shown in Fig. 6.

As shown in Fig. 6, QP has little effect on the hit ratio, and the hit ratio increases as the Ths *b*, *c*, *d*, and *e* increase. The bigger Th value can achieve a higher hit ratio with small RD loss, but the reduction of computing complexity will be smaller, so the selection of the Th value should consider the trade-off between RD loss and computing complexity reduction. According to the testing results shown in Fig. 6, we set the target hit ratio as 95%, so we can select the Ths *b*–*e* as 6, 2, 1.5, and 2, respectively.

In this study, AG_{max} (the maximum value of AGH, AGV, AGMD, and AGDD) of CUs, which perform the partition, are investigated under different QPs and depths. The results show that AG_{max} is generally larger when the CU is split. So a Th can be set to determine whether the CU is a uniform CU or not. When AG_{max} is smaller than Th, the CU is considered to be uniform. Th is determined using the hit ratio of CU split, and the target hit ratio is set to 95%, and Th values under different QPs and depths are shown in Fig. 7. After Th is determined, Th *a* can be set as $a = 0.3 \times \text{Th}$.

3.3 Prediction mode selection fast algorithm flow

The modified 'three-step intra prediction mode fast selection' strategy includes RMD process, MPM acquisition process and RD

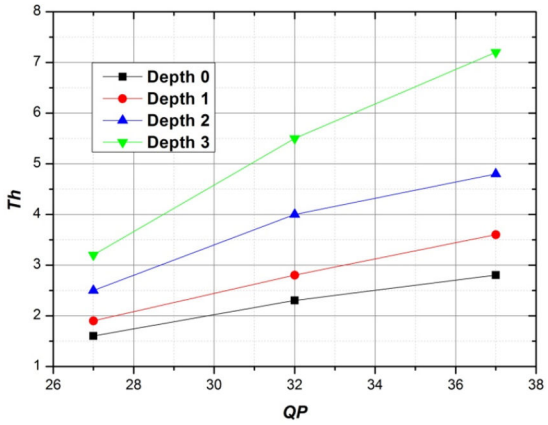

Fig. 7 *Relationship between Th and depth, QP*

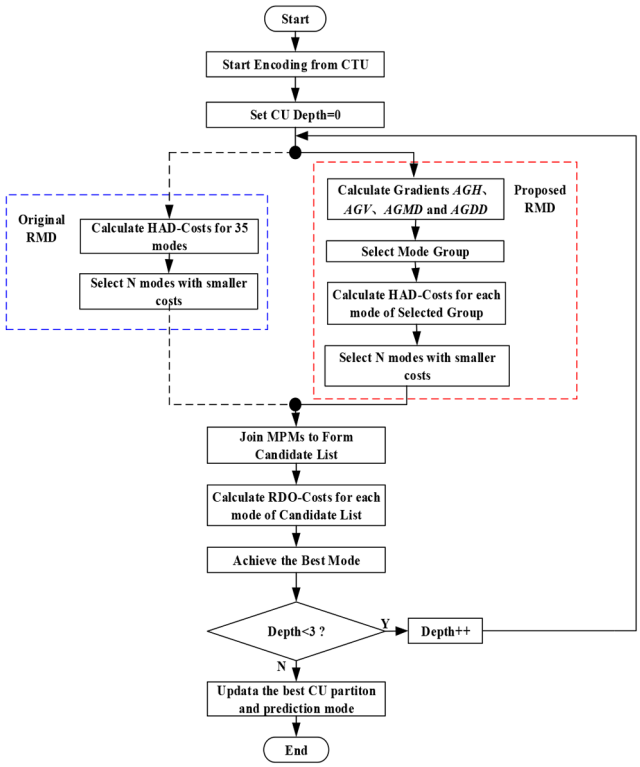

Fig. 8 *Flowchart of intra prediction mode selection fast algorithm*

optimisation (RDO) process as shown in Fig. 8. In the original strategy, the HAD costs of 35 prediction modes should be calculated during the RMD process. The proposed fast algorithm for intra prediction mode selection based on mode grouping is to add preprocessing in RMD process. First the average gradients AGH, AGV, AGMD, and AGDD of PU are calculated, and then the corresponding mode group is selected according to the ratio of the average gradients, so only the HAD costs of the modes included in the mode group are calculated. The modes with smaller HAD-costs are selected and are added to the MPM to form a candidate list of RDO, then the best intra prediction mode is determined by the RDO process. The fast algorithm can reduce the number of modes entering the RMD process, and effectively reduce the computing complexity of intra prediction mode selection.

4 Intra CU size selection fast algorithm

HEVC uses a 64×64 LCU quadtree recursive partitioning mechanism to determine the CU size. Compared with a fixed macroblock size of 16×16 in H.264/AVC, this mechanism can reduce the bit rate by about 12% at the cost of multiple computing complexity. In this study, the texture distribution feature of CU is considered, and the CU size decision mechanism based on multiple textures is established, which realises the fast selection of CU size.

4.1 Relationship between CU size and content

By analysing the relationship between CU partitioning and video content in intra prediction, it can be seen that when the optimal CU size is 64×64 and 32×32, it is basically a region with flat and uniform texture. When the optimal CU size is 16×16 and 8×8, it is basically a region with rich and complex texture. Therefore, we can utilise texture features of the image to improve the traversal quadtree partitioning process, to further improve the intra coding efficiency of HEVC.

4.2 Extraction of multiple texture features

4.2.1 Content complexity: Through the above analysis, whether the current CU performs quadtree partitioning depends mainly on the content complexity of the CU. To accurately reflect the content complexity of CU, this study employs neighbourhood mean square error (NMSE) of CU, as shown in (9) and (10)

$$\bar{p}(i,j) = \begin{cases} 0, & i = 0 \text{ or } N-1 \\ 0, & j = 0 \text{ or } N-1 \\ \dfrac{1}{8}\begin{pmatrix} p(i-1,j-1)+p(i-1,j) \\ +p(i-1,j+1)+p(i,j-1) \\ +p(i,j+1)+p(i+1,j-1) \\ +p(i+1,j)+p(i+1,j+1) \end{pmatrix}, & \text{others} \end{cases} \tag{9}$$

$$\text{NMSE} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(p(i,j)-\bar{p}(i,j)\right)^2 \tag{10}$$

where N represents the size of CU, $p(i, j)$ represents the luminance value of a pixel located at (i, j) in CU, and $\bar{p}(i, j)$ represents the mean of the eight-neighbour luminance values of the current pixel. The smaller NMSE means the current CU is more uniform.

4.2.2 Direction complexity: Apart from content complexity, direction information is also an important parameter for measuring texture complexity. This study defines the average gradient sum (SAG) to represent the direction information of a CU as

$$\text{SAG} = \text{AGH} + \text{AGV} + \text{AGMD} + \text{AGDD} \tag{11}$$

The larger SAG means a larger average gradient of current CU.

4.2.3 Complexity differences between sub-CUs: According to the analysis in Section 4.1, it can be found that the quadtree partition of parent CU is not always a full quadtree. In most cases, flat CU generally does not perform CU partitioning. For example, for a 64×64 CU, whether to split into four 32×32 sub-CUs depends on the difference of texture complexity between the four sub-CUs. Even if the 64×64 CU's texture complexity is low, when the texture complexity difference between the four sub-CUs is large, then the 64×64 CU still needs to perform quadtree partitioning. To accurately reflect the difference between the texture complexity of the four sub-CUs, this study defines the NMSE difference (Sub_NMSED) and the difference of SAG (Sub_SAGD) of the sub-CUs, as

$$\begin{aligned} \text{Sub\_NMSED} &= \frac{1}{4}\sum_{i=0}^{3}\left(\text{NMSE}_i - \overline{\text{NMSE}}\right)^2 \\ \text{Sub\_SAGD} &= \frac{1}{4}\sum_{i=0}^{3}\left(\text{SAG}_i - \overline{\text{SAG}}\right)^2 \end{aligned} \tag{12}$$

where NMSE_i and SAG_i are the NMSE and SAG of the four sub-CUs respectively, $\overline{\text{NMSE}}$ and $\overline{\text{SAG}}$ are the mean value of NMSE_i and SAG_i, respectively.

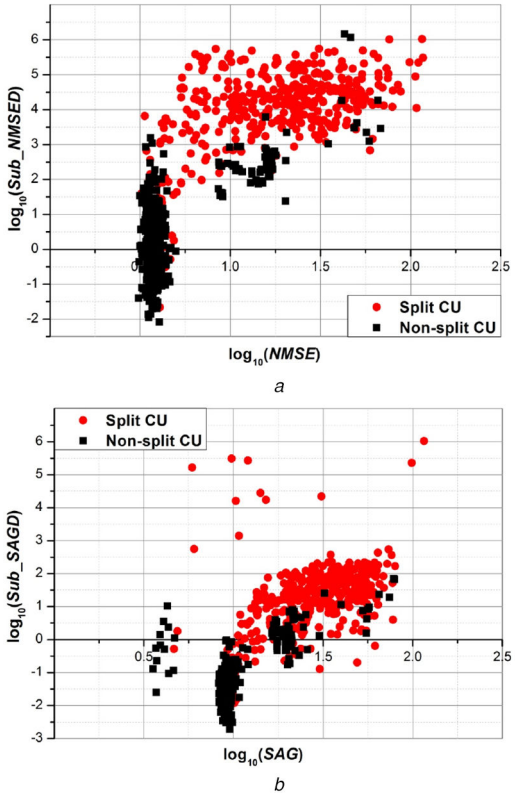

Fig. 9 *Distribution of multiple textures of CUs*
(a) Sub_NMSED versus NMSE, ***(b)*** Sub_SAGD versus SAG

Table 2 Decision strategy of CU classification

$f^{A}_{Depth}(\boldsymbol{x}) < 0$ & $f^{B}_{Depth}(\boldsymbol{x}) < 0$	Group A
$f^{B}_{Depth}(\boldsymbol{x}) > 0$ & $f^{A}_{Depth}(\boldsymbol{x}) > 0$	Group B
$f^{A}_{Depth}(\boldsymbol{x}) < 0$ & $f^{B}_{Depth}(\boldsymbol{x}) > 0$	Group C
$f^{B}_{Depth}(\boldsymbol{x}) > 0$ & $f^{A}_{Depth}(\boldsymbol{x}) < 0$	

We use ×265 to encode the first ten frames of ParkScene with QP = 32 and obtain multiple textures of 64 × 64 CUs. In Fig. 9*a*, *X*-axis is $\log_{10}$NMSE, *Y*-axis is $\log_{10}$Sub_NMSED, red dot indicates that the 64 × 64 CU continues to be split into four 32 × 32 CUs, and the black dot indicates that 64 × 64 CU is no longer split. Similarly, Fig. 9*b* reflects the distribution of $\log_{10}$SAG and $\log_{10}$Sub_SAGD. It can be seen that except for some outliers, the red dots and black dots are concentrated, which indicates that the multiple textures NMSE, SAG, Sub_NMSED and Sub_SAGD are proper features for characterising whether the CU needs to be split.

4.3 Sample calibration and model training

We use class A to represent CUs that need to be split (split CU) and class B to represent CUs that need not to be split (non-split CU). According to the statistics result, classes A and B are partially overlapped and cannot be separated linearly depending on the features of NMSE, SAG, Sub_NMSED, and Sub_SAGD. So this study proposes a two-SVM combination scheme to classify the CU. Each SVM defines a decision function as

$$f(\boldsymbol{x}) = \text{sign}\left(\left(\sum_{i=1}^{M} \alpha_i^* y_i \boldsymbol{x}_i^{\mathrm{T}}\right) \cdot \boldsymbol{x} + b\right) \tag{13}$$

where $x_i = [\text{logNMSE}, \text{logSAG}, \text{logSub}_{\text{NMSED}}, \text{logSub}_{\text{SAGD}}]^{\mathrm{T}}$, log denotes $\log_{10}$, $y_i \in \{+1, -1\}$ +1 denotes class A, −1 denotes class B; α_i is the Lagrange factor, b is the offset, M denotes the number of training samples.

The above four video sequences are used to train the SVM, and we can get the decision functions of classes A and B under each depth. We call the CU that can be separated by two SVMs as groups A and B, and the CU that cannot be classified into the two groups as group C. The decision functions of classes A and B are represented as $f^{A}_{Depth}(\boldsymbol{x})$ and $f^{B}_{Depth}(\boldsymbol{x})$ under different depth, respectively, and the decision strategy of CU classification is shown in Table 2.

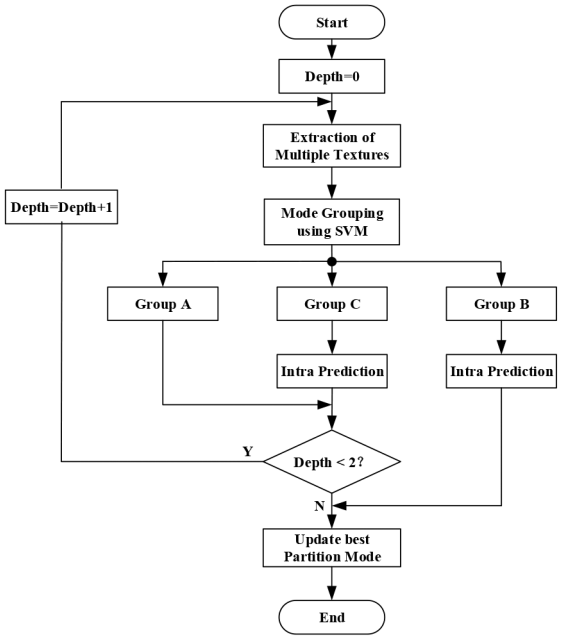

Fig. 10 *Flowchart of CU size selection fast algorithm*

4.4 CU size selection fast algorithm flow

The determination of CU's partition mode according to multiple textures by the combination of two SVMs can accelerate the intra CU quadtree partitioning process. The intra CU size selection fast algorithm flowchart is shown in Fig. 10.

First, the multi-texture features of current CU are extracted, and then use $f^{A}_{Depth}(\boldsymbol{x})$ and $f^{B}_{Depth}(\boldsymbol{x})$ to determine the class of CU according to the extracted features. If the decision is group A, it indicates that CU's texture is complex, the intra prediction process can be skipped, and split into sub-CUs directly, which is called strategy 'early split'. If the decision is group B, it indicates that current CU is flat enough to terminate quadtree partition ahead. After completing the intra prediction of current CU, optimal CU partition mode is updated, which is called strategy 'early termination'. If the decision is group C, it indicates that the algorithm cannot make a fast decision according to its texture features, and current CU performs intra-prediction and then split into sub-CUs according to the original algorithm of x265. After current LCU traverses all depths, the optimal partition mode of LCU is updated.

By adopting the early termination strategy on regions of flat content, the time of subsequent sub-CUs' intra prediction can be saved. Adopting the early split strategy on regions of complex content can save current CU's intra prediction time. The combination of two SVMs can greatly reduce the misjudgment of CUs with insignificant texture features, and achieve a good balance between saving coding time and maintaining RD performance.

5 Experimental results and analysis

Our research focuses on embedded system implementation of HEVC. Open source encoder x265 offers the highest compression efficiency and performance on a wide variety of hardware platforms, so we choose ×265 as the reference encoding software [31].

The hardware platform is Intel Core i7-4790, the clock speed is 3.6 GHz, the memory is 8 GB, and the operating system is Windows 7 64-bit. The reference software is ×265-1.7. The

Table 3 CU classification result of test samples

Function	Depth	Class	Precision, %			Recall, %		
			QP = 27	QP = 32	QP = 37	QP = 27	QP = 32	QP = 37
$f^{A}_{Depth}(\boldsymbol{x})$	0	A	98.52	98.62	98.85	95.00	95.47	96.40
	1	A	98.63	98.79	98.05	79.40	65.44	58.93
	2	A	98.56	98.47	98.17	27.97	34.20	30.21
$f^{B}_{Depth}(\boldsymbol{x})$	0	B	98.33	98.04	97.63	84.29	93.33	92.54
	1	B	98.40	98.24	97.16	70.38	74.95	71.93
	2	B	97.83	97.81	98.41	68.22	71.40	70.09

Table 4 Experimental results of the proposed fast algorithms

Resolution	Video	Algorithm 1			Algorithm 2			Algorithm 1 + 2		
		ΔTime	ΔBitrate	$\Delta PSNR_Y$	ΔTime	ΔBitrate	$\Delta PSNR_Y$	ΔTime	ΔBitrate	$\Delta PSNR_Y$
2560 × 1600	PeopleOnStreet	8.68	0.70	0.008	32.83	0.29	0.012	40.03	1.00	0.020
	Traffic	12.40	0.34	0.007	35.80	0.43	0.008	43.92	0.91	0.010
1920 × 1080	BasketballDrive	8.63	0.20	0.001	39.89	1.36	0.049	44.85	1.58	0.058
	BQTerrace	11.54	0.42	0.003	38.97	0.53	−0.008	45.45	0.89	−0.003
	Cactus	5.16	0.21	0.002	36.55	0.60	0.004	39.53	0.81	0.007
	ParkScene	4.48	0.05	0.003	37.31	0.44	0.004	40.55	0.51	0.007
832 × 480	Basketballdrill	7.11	0.34	0.003	32.31	0.17	0.016	38.38	0.52	0.020
	BQMall	10.15	0.44	0.008	34.66	0.74	0.011	42.21	1.21	0.018
	PartyScene	5.73	0.33	0.007	32.30	0.11	0.001	37.04	0.44	0.009
	RaceHorses	8.43	0.12	0.003	32.61	0.48	−0.009	38.75	0.62	−0.006
416 × 240	BasketballPass	11.24	0.49	0.005	31.51	0.04	0.040	40.09	0.73	0.044
	BlowingBubbles	6.45	0.16	0.006	29.99	0.09	−0.002	34.48	0.21	0.002
	BQSquare	5.28	0.70	0.015	32.22	0.03	0.001	35.47	0.71	0.018
1280 × 720	FourPeople	15.63	0.72	0.007	42.49	0.46	0.011	51.13	1.20	0.018
	Johnny	14.58	0.93	0.006	52.41	1.54	0.039	58.20	2.36	0.043
	KristenAndSara	15.61	0.97	0.007	47.51	0.99	0.019	54.76	1.98	0.028
Average		9.44	0.45	0.006	36.84	0.52	0.012	42.80	0.98	0.018

encoding configuration is all-intra mode, without assembly acceleration and multithread support. The experiment uses the standard test video set released by JCT-VC [32]. Each video sequence encodes 100 frames.

5.1 Classification result of CU split

The accuracy of the two-SVM classifier can greatly affect the performance of the CU size selection fast algorithm. In this study, we use two parameters Precision and Recall to measure the performance of the classifier, as

$$\begin{aligned} \text{Precision(I)} &= \frac{\text{TP(I)}}{\text{TP(I)} + \text{TF(I)}} \\ \text{Recall(I)} &= \frac{\text{TP(I)}}{\text{TP(I)} + \text{FN(I)}} \end{aligned} \tag{14}$$

where TP(I) denotes the number of samples in class I test set which are correctly judged to be class I, TF(I) denotes the number of samples not in class I that are misjudged to be class I, FN(I) denotes the number of samples in class I that are misjudged to be other classes.

Table 3 gives the Precision and Recall of class A and class B classification under different QPs and depths. To reduce the distortion of intra coding, Precision of both types are controlled at about 98% when training the SVM model. As shown in Table 3, it can be found that Recall is highest when depth = 0, indicating that the model has good performance on 64 × 64 CUs. Moreover, Recall is relatively high when depth = 1, lowest when depth = 2, indicating that the model has a worse performance on 32 × 32 and 16 × 16 CUs. It also reflects that the larger CUs tend to have a larger difference in textures between split and non-split.

5.2 Performance analysis

This paper mainly uses ΔTime, ΔBitrate, and $\Delta PSNR_Y$ to measure the performance of the proposed fast algorithms

$$\begin{aligned} \Delta\text{Time} &= \frac{\text{Time}_{\text{x265}} - \text{Time}_{\text{proposed}}}{\text{Time}_{\text{x265}}} \times 100\% \\ \Delta\text{Bitrate} &= \frac{\text{Bitrate}_{\text{proposed}} - \text{Bitrate}_{\text{x265}}}{\text{Bitrate}_{\text{x265}}} \times 100\% \\ \Delta\text{PSNR}_Y &= \text{PSNR}_{\text{x265, }Y} - \text{PSNR}_{\text{proposed, }Y} \end{aligned} \tag{15}$$

where $\text{Time}_{\text{x265}}$, $\text{Bitrate}_{\text{x265}}$ and $\text{PSNR}_{\text{x265, }Y}$ are encoding time, bit rate, and luminance peak-signal-to-noise ratio (PSNR) of ×265-1.7, respectively. $\text{Time}_{\text{proposed}}$, $\text{Bitrate}_{\text{proposed}}$, and $\text{PSNR}_{\text{proposed, }Y}$ are the proposed fast algorithm's coding time, bit rate and luminance PSNR, respectively, and QP set to 27, 32, 37. ΔBitrate and $\Delta PSNR_Y$ reflect the RD performance of the proposed algorithms, ΔTime reflects the coding time saved by the proposed fast algorithms. The testing results are shown in Table 4 (Algorithm 1 denotes the proposed prediction mode selection fast algorithm, Algorithm 2 denotes the proposed CU size selection fast algorithm).

It can be seen from Table 4 that the proposed intra prediction mode selection fast algorithm based on mode grouping saves the coding time by 9.44% averagely, while drops PSNR by 0.006 dB and increases bit rate by 0.45% compared with the ×265-1.7 original algorithm. The proposed algorithm can effectively save the coding time for the 720P video sequences. This is because the 720P video sequences used in the test have the feature of complex foreground and monotonous background. For large flat background PUs, they basically select mode group 1 (DC mode and planar mode) to enter the RMD process. For complex video sequences, such as PartyScene, BQSquare, and Catus, encoding time is not significantly reduced, leading to a larger bit rate increase instead.

Table 5 Performance comparison of different algorithms for the sequences

Resolution	Sequences	Ours BDBR (%)/ΔT (%)	Ref. [16] BDBR (%)/ΔT (%)	Ref. [28] BDBR (%)/ΔT (%)	Ref. [29] BDBR (%)/ΔT (%)
2560 × 1600	Traffic	0.66/46.28	1.40/51.00	0.98/45.69	1.44/49.10
1920 × 1080	BasketballDrive	1.32/59.80	1.50/66.00	1.87/61.09	2.36/49.70
	ParkScene	0.72/48.30	0.90/51.00	0.67/40.01	1.00/47.30
832 × 480	Basketballdrill	1.49/42.20	1.50/44.00	0.99/39.74	0.88/48.20
1280 × 720	Johnny	1.45/62.90	2.00/61.00	3.01/67.99	2.23/49.70
	KristenAndSara	1.17/59.20	1.90/61.00	2.39/63.56	2.21/49.30
	Average	1.13/53.11	1.53/55.67	1.65/53.01	1.69/48.88

The proposed algorithm has different encoding acceleration effects for video sequences with different texture features, because video content directly affects the decision of mode group, resulting in a large difference in the number of modes entering the RMD process.

It can also be seen from Table 4 that the proposed CU size selection fast algorithm based on multi-texture saves the coding time by 36.84% averagely, while increases the bit rate by 0.52% and drops PSNR by 0.012 dB compared with ×265-1.7. The proposed algorithm has good coding acceleration effects on various video sequences. This is because the early split strategy is adopted for CUs with complex textures, and the early termination strategy is adopted for CUs with flat textures. The complement of the two strategies effectively reduces the computing complexity of the CU size selection process.

Compared with the ×265-1.7 original algorithm, combining the two fast algorithms mentioned above saves the coding time by 42.80%, increases the bit rate by 0.98% and drops the PSNR of luminance component only by 0.018 dB.

To verify the effectiveness of the proposed algorithm, we also implemented the fast algorithms on HEVC test model (HM) and tested the performance. All frames were encoded under the all-intra (AI) mode and four QPs (22, 27, 32 and 37). The CTU size is 64 × 64. Bjontegaard delta bitrate (BDBR) is used to represent the average bit rate loss [33], and ΔT is used to represent coding time-saving in percentage (same as ΔTime).

We compared our method with three studies and list the result in Table 5. Our method is tested on HM16.20, and [16, 28, 29] are on HM12.0, HM16.7, and HM15, respectively. From Table 5, it can be seen that the method proposed in [16] saves a little more time than our method with more bit rate loss. We also tested that under the AI mode, the encoding time of HM12.0 is nearly double of its of HM16.20, so HM12.0 is easier to be optimised in general. In [28], the acceleration result is nearly the same as our method, but the bit rate loss is relatively more. In [29], the performance of time-saving and bit rate loss is both worse than our method. In a word, our proposed algorithms have achieved a good balance between time reduction and bit rate loss. Furthermore, the Bjontegaard delta peak signal-to-noise ratio (BDPSNR) of [28, 29] is −0.07 dB, the BDPSNR of our method is −0.043 dB.

As we know, VP9 and HEVC are two common video codecs. HM is a test model of HEVC, which only offers the prototype implementation of HEVC. X265 is a high-speed HEVC encoder, VP9 is an open-source codec developed by Google. Testing results show that under the AI mode, the encoding time of VP9 and X265-1.7 are about 10 and 1% of its of HM16.20, respectively.

On our PC, X265-1.7 needs 0.7–0.9 s to encode one frame of high-definition video under intra coding mode. Using our proposed fast algorithm, the encoding time can be reduced to 0.4–0.5 s, while the PSNR is nearly unchanged and the bit rate is only increased a little, which can offer a better user experience.

6 Conclusion

To reduce the computing complexity of intra prediction mode selection, this study proposes a fast algorithm for intra prediction mode selection based on mode grouping. According to the statistics of best intra prediction mode, 35 intra prediction modes are grouped, and mode groups are selected by the gradient values of PU in horizontal, vertical, 45° and 135° directions and only the modes in the selected mode group can enter the RMD process. To reduce the computing complexity of quadtree partition of intra CU, this study proposes a fast algorithm for intra CU size selection based on multiple textures. Based on the statistical analysis of the relationship between CU texture features and CU size, content complexity, direction complexity, content complexity difference, and direction complexity difference of sub-CUs are defined to characterise the texture features of CU. Then, based on the extracted features, two SVMs are used to make decision of CU size, and CUs with complex textures are split in advance, CUs with simple textures terminate split early. The two algorithms mentioned above are combined to perform coding performance testing. Experimental results show that the two fast algorithms proposed in this study can not only save the intra coding time effectively but also ensure the video quality with little bit rate increase.

7 References

[1] ITU-T Recommendation H.265.: 'High efficiency video coding', International Telecommunication Union, 2013

[2] Sullivan, G.J., Ohm, J.R., Han, W.J., *et al.*: 'Overview of the high efficiency video coding (HEVC) standard', *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**, (12), pp. 1649–1668

[3] Ohm, J.R., Sullivan, G.J., Schwarz, H., *et al.*: 'Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC)', *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**, (12), pp. 1669–1684

[4] Tan, T.K., Weerakkody, R., Mrak, M., *et al.*: 'Video quality evaluation methodology and verification testing of HEVC compression performance', *IEEE Trans. Circuits Syst. Video Technol.*, 2016, **26**, (1), pp. 76–90

[5] Jiang, W., Chi, Y., Jin, H., *et al.*: 'A fine-grained parallel intra prediction for HEVC based on GPU'. IEEE Int. Conf. on Parallel and Distributed Systems, Wuhan, 2016, pp. 778–784

[6] Kim, J., Choe, Y., Kim, Y.G.: 'Fast coding unit size decision algorithm for intra coding in HEVC'. IEEE Int. Conf. on Consumer Electronics, Las Vegas, 2013, pp. 637–638

[7] Chen, J., Yu, L.: 'Effective HEVC intra coding unit size decision based on online progressive Bayesian classification'. IEEE Int. Conf. on Multimedia and Expo, Seattle, 2016, pp. 1–6

[8] Zhang, T., Sun, M., Zhao, D., *et al.*: 'Fast intra-mode and CU size decision for HEVC', *IEEE Trans. Circuits Syst. Video Technol.*, 2017, **27**, (8), pp. 1714–1726

[9] Piao, Y., Min, J., Chen, J., *et al.*: 'Encoder improvement of unified intra prediction'. JCTVC-C207, Guangzhou, October 2010

[10] Zhao, L., Zhang, L., Ma, S., *et al.*: 'Fast mode decision algorithm for intra prediction in HEVC'. IEEE, Visual Communications and Image Processing, Tainan, 2011, pp. 1–4

[11] Jiang, W., Ma, H., Chen, Y.: 'Gradient based fast mode decision algorithm for intra prediction in HEVC'. Int. Conf. on Consumer Electronics, Communications and Networks (CECNet), Yichang, 2012, pp. 1836–1840

[12] Abdelrasoul, M., Sayed, M.S., Goulart, V.: 'Diagonal-based fast intra-mode decision algorithm for HEVC', *IET Image Process.*, 2017, **11**, (10), pp. 888–898

[13] Chen, G., Pei, Z., Sun, L., *et al.*: 'Fast intra prediction for HEVC based on pixel gradient statistics and mode refinement'. IEEE China Summit & Int. Conf. on Signal and Information Processing, Beijing, 2013, pp. 514–517

[14] Zhang, H., Ma, Z.: 'Fast intra mode decision for high efficiency video coding (HEVC)', *IEEE Trans. Circuits Syst. Video Technol.*, 2014, **24**, (4), pp. 660–668

[15] Jamali, M., Coulombe, S., Caron, F.: 'Fast HEVC intra mode decision based on edge detection and SATD costs classification'. Data Compression Conf., Snowbird, UT, 2015, pp. 43–52

[16] Tseng, C.F., Lai, Y.T.: 'Fast coding unit decision and mode selection for intra-frame coding in high-efficiency video coding', *IET Image Process.*, 2016, **10**, (3), pp. 215–221

[17] Reuze, K., Philippe, P., Deforges, O., *et al.*: 'Intra prediction modes signalling in HEVC'. Picture Coding Symp. (PCS), Nuremberg, 2016, pp. 1–5

[18] Zhu, W., Yi, Y., Zhang, H., *et al.*: 'Fast mode decision algorithm for HEVC intra coding based on texture partition and direction', *J. Real-Time Image Process.*, 2020, **17**, pp. 275–292, https://doi.org/10.1007/s11554-018-0766-z

[19] Shi, Y., Au, O.C., Zhang, X., *et al.*: 'Content based fast prediction unit quadtree depth decision algorithm for HEVC'. IEEE Int. Symp. on Circuits and Systems (ISCAS2013), Beijing, 2013, pp. 225–228
[20] Wang, Y., Fan, X., Zhao, L., *et al.*: 'A fast intra coding algorithm for HEVC'. IEEE Int. Conf. on Image Processing (ICIP), Paris, 2014, pp. 4117–4121
[21] Sun, X., Chen, X., Xu, Y., *et al.*: 'Fast CU partition strategy for HEVC based on Haar wavelet', *IET Image Process.*, 2017, **11**, (9), pp. 717–723
[22] Öztekin, A., Erçelebi, E.: 'An early split and skip algorithm for fast intra CU selection in HEVC', *J. Real-Time Image Process.*, 2016, **12**, (2), pp. 273–283
[23] Cho, S., Kim, M.: 'Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding', *IEEE Trans. Circuits Syst. Video Technol.*, 2013, **23**, (9), pp. 1555–1564
[24] Lim, K., Lee, J., Kim, S., *et al.*: 'Fast PU skip and split termination algorithm for HEVC intra prediction', *IEEE Trans. Circuits Syst. Video Technol.*, 2015, **25**, (8), pp. 1335–1346
[25] Hsu, H., Huang, S., Lin, Y.: 'Computational complexity reduction for HEVC intra prediction with SVM'. IEEE Global Conf. on Consumer Electronics (GCCE), Nagoya, 2017, pp. 1–2
[26] Katayama, T., Kuroda, K., Shi, W., *et al.*: 'Low-complexity intra coding algorithm based on convolutional neural network for HEVC'. Int. Conf. on Information and Computer Technologies (ICICT), DeKalb, IL, 2018, pp. 115–118
[27] Feng, Z., Liu, P., Jia, K., *et al.*: 'HEVC fast intra coding based CTU depth range prediction'. IEEE Int. Conf. on Image, Vision and Computing (ICIVC), Chongqing, 2018, pp. 551–555
[28] Zhang, Y., Pan, Z., Li, N., *et al.*: 'Effective data driven coding unit size decision approaches for HEVC INTRA coding', *IEEE Trans. Circuits Syst. Video Technol.*, 2018, **28**, (11), pp. 3208–3222
[29] Jamali, M., Coulombe, S.: 'Fast HEVC intra mode decision based on RDO cost prediction', *IEEE Trans. Broadcast.*, 2019, **65**, (1), pp. 109–122
[30] Shen, L., Zhang, Z., An, P.: 'Fast CU size decision and mode decision algorithm for HEVC intra coding', *IEEE Trans. Consum. Electron.*, 2013, **59**, (1), pp. 207–213
[31] 'X265 HEVC encoder/H.265 video codec', Available at: http://www.x265.org/, accessed 26 February 2019
[32] Bossen, F.: 'Common test conditions', JCTVC-H1100, March 2012
[33] Bjøntegaard, G.: 'Calculation of average PSNR differences between RD curves', VCEG-M33, April 2001