Zhang et al. EURASIP Journal on Image and Video Processing (2017) 2017:86
DOI 10.1186/s13640-017-0237-7

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Fast and adaptive mode decision and CU partition early termination algorithm for intra-prediction in HEVC

Mengmeng Zhang*, Xiaojun Zhai and Zhi Liu

Abstract

High Efficiency Video Coding (HEVC or H.265), the latest international video coding standard, displays a 50% bit rate reduction with nearly equal quality and dramatically higher coding complexity compared with H.264. Unlike other fast algorithms, we first propose an algorithm that combines the CU coding bits with the reduction of unnecessary intra-prediction modes to decrease computational complexity. In this study, we first analyzed the statistical relationship between the best mode and the costs calculated through Rough Mode Decision (RMD) process and proposed an effective mode decision algorithm in intra-mode prediction process. We alleviated the computation difficulty by carrying out the RMD process in two stages, reducing 35 modes down to 11 modes in the first RMD process stage, and adding modes adjacent to the most promising modes selected during the first stage into the second RMD stage. After these two stages, we had two or three modes ready to be used in the rate distortion operation (RDO) process instead of the three or eight in the original HEVC process, which significantly reduced the number of unnecessary candidate modes in the RDO process. We then used the coding bits of the current coding unit (CU) as the main basis for judging its complexity and proposed an early termination method for CU partition based on the number of coding bits of the current CU. Experimental results show that the proposed fast algorithm provides an average time reduction rate of 53% compared to the reference HM-16.12, with only 1.7% Bjontegaard delta rate increase, which is acceptable for Rate-Distortion performance.

Keywords: HEVC, CU partition, Coding bits, Mode, Fast algorithm

1 Introduction

High Efficiency Video Coding (HEVC) is the newest video coding standard developed by the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations and the Joint Collaborative Team on Video Coding (JCT-VC) [1]. HEVC aims to achieve an efficiently high coding compression rate compared to H.264/AVC [2], especially with high-resolution video content. HEVC still uses the hybrid coding framework based on motion compensation, which was adapted by H.261. Under this framework, inter-frame prediction is used to eliminate the correlation of the domains of time and space. The prediction residual uses discrete cosine transform and quantization to eliminate spatial correlation. Adaptive entropy coding eliminates statistical redundancy. A loop filter is used to eliminate the quantization noise, but the HEVC has caused significant improvements in such aspects as the loop filter, image coding unit, context-adaptive binary arithmetic coding (CABAC), directional intra-prediction, advanced motion vector prediction and merge, subpixel motion estimation and compensation, and sample adaptive offsets (SAO). Using object-oriented metrics, HEVC intra-coding achieves an average bitrate reduction of 22% and up to 36% over H.264/AVC [3]. Compared with H.264, HEVC supports a larger range of coding block sizes and adapts a more flexible quad-tree coding unit to solve problems in video images, including having different colors and textures, reference frame correlation, and partial information. According to the different functions, HEVC consists of four feature coding block units, including coding tree unit (CTU), coding unit (CU), prediction unit (PU), and transform unit (TU). In the current HEVC test model (HM) [4],

* Correspondence: muchmeng@126.com
North China University of Technology, Beijing, China

video images are first divided into slices, which are then divided into CTUs of equal size or largest coding unit (LCU). An LCU is an $N \times N$ (64×64) block of luma simples paired with two corresponding blocks of chroma samples, the concept of which is broadly analogous to that of a macroblock (MB) in previous standards such as H.264/AVC [5]. The CTU allows a quad-tree to be split recursively into four CUs of equal size. Each CU can then be encoded or split into four sub-CUs of equal size, and so on, with the process ending only when the smallest CU is reached. Splitting is the reason for the variation in the CU's size and depth from 64×64 to 8×8 and from 0 to 3, respectively. Figure 1 shows the quad-tree structure formed by CUs in HEVC. An LCU comprises optimal CU partitions is used as a low bit rate to encode the picture content diversity adaptively. These optimal CUs have the smallest rate distortion (RD) cost generated by a rate distortion optimization (RDO) process. When predicting a CU, it may be divided into PUs that contain individual prediction information. PU sizes and depths range from 64×64 to 4×4 and from 0 to 4, respectively. Each PU has up to 35 prediction modes, including a PLANAR, a DC, and 33 directional modes as shown in Fig. 2. The DC and PLANAR are two non-directional modes applied to predict the area with homogeneous content. The 33 directional modes improve prediction accuracy, but adding computation difficulty significantly.

Figure 3 shows a picture divided into optimal CUs in HEVC. The picture sufficiently proves that the smaller CUs contain more information and texture complexity. In contrast, larger CUs contain lesser information. The HEVC encoder has to search for all possible CUs to obtain optimal CUs, resulting in an extremely large computation [6]. The process of finding the optimal CUs is highly time-consuming because the final optimal CU can either be the current CU or any of the four sub-CUs, depending on whether the RD cost of the current CU is greater or lesser than the sum of RD costs of the four sub-CUs. Thus, the encoder has to encode CUs

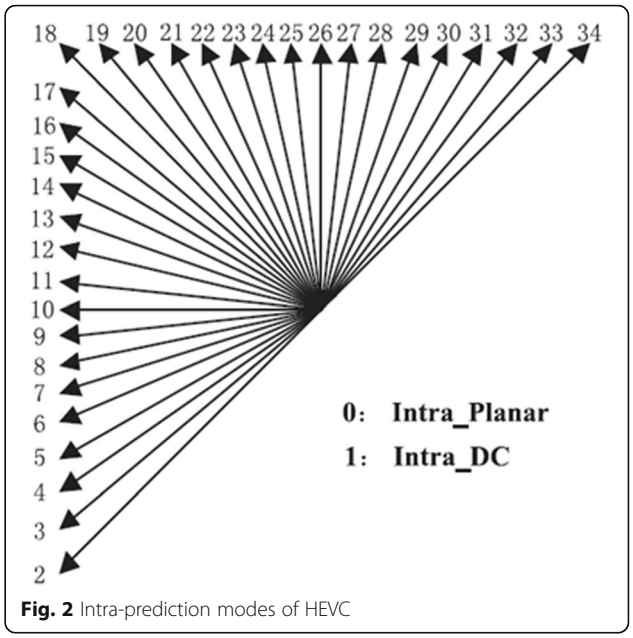

Fig. 2 Intra-prediction modes of HEVC

with sizes 64×64, 32×32, 16×16, and 8×8. An LCU measuring 64×64 is divided into four CUs measuring 32×32 each, which are then divided into four 16×16 CUs, with every 16×16 CU divided into four 8×8 CUs, yielding a total of 85 CUs for each LCU. Hence, 35 prediction modes have to be tested under each of the 85 CUs to gain the optimal partition plan. The proposed method seeks to decrease the 35 prediction modes to obtain the optimal CU partition as soon as possible. The prediction accuracy is equal to that of HEVC standards, thereby reducing the burden of computation. Finding a new algorithm to solve the computation difficulty is necessary. Hence, to widen the range of applications, HEVC should be widened to ultra-high-definition formats known as 4K and 8K and real-time application scenarios.

1.1 Related work

Since the emergence of HEVC, several researchers and scientific research institutions focused considerable

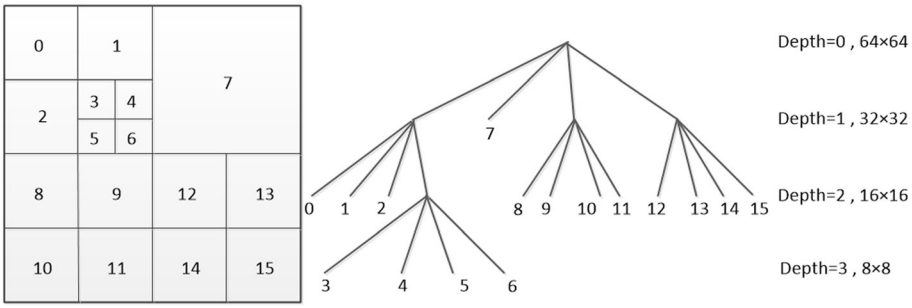

Fig. 1 The quad-tree structure of an LCU partition. To the left is an LCU partition showing CUs sized 64×64 to 8×8. To the right is the corresponding quad-tree structure together with depths and CU sizes

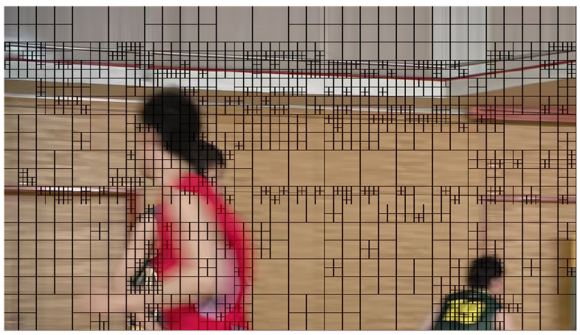

Fig. 3 The optimal CUs of HEVC

efforts in increasing the efficiency of HEVC, including carrying out various approaches to reduce intra-prediction computation difficulty and coding complexity. There are many other researches aimed to solve video coding problems besides HEVC, for example [7–10], these papers also contributed to the development of video coding. But in this paper, we mainly optimize HEVC encoding. Ahn et al. [11] parallelized HEVC encoder interpolation filter, cost function, and transform to reduce the complexity of HEVC, which is caused by single instruction, multiple data (SIMD) operations, and data-level parallelism. Min et al. [12] proposed a distributed video coding method with a hierarchical group of picture structure. Yan et al. [13] proposed a parallel framework to decouple motion estimation for different partitions on many-core process. The parallel deblocking filter for HEVC was proposed by Yan et al. [14]; they decrease HEVC complexity in decoder process. Peng et al. [15] introduced a fast coding algorithm that uses depth and color features to extract depth discontinuous regions, depth edge regions, and motion regions as masks for efficient processing and fast coding. Nishikori et al. [16] of the Department of Information and Electronics of the Tottori University in Japan proposed a fast CU size selection method that determined CU sizes using the variance value of the input image. Bai [17] put forward a fast coding tree unit algorithm that utilizes Sobel gradient and mean absolute deviation values to analyze the texture of the CU while filtering out unnecessary CU candidates to speed up the original intra-coding in HEVC. Blasi et al. [18] proposed a method that involves visiting the smallest CUs first and continuing with the larger CUs up the quad-tree and then extracting useful information to decide if a CTU is encoded using the reverse CU visiting. Belghith et al. [19] proposed a CU partition algorithm based on the Sobel edge detection process to decide on the appropriate CU size early. Goswami et al. [20] proposed a new approach that utilizes the RD costs of the parent and current levels to terminate the quad-tree-based structure early, saving average time of 38.03%, 1.3% BD-rate increase compared with HM 10.0. Shen et al. [21] proposed a CU-splitting, early termination algorithm that makes use of a support vector machine (SVM). Cen et al. [22] introduced a fast adaptive CU depth decision mechanism that utilizes the special correlations in the sequence frame, saving average time of 16% compared with HM 10.0. The aforementioned papers [15–22] are fast algorithms aimed at speeding up the HEVC intra-prediction encoder through early termination of CU partition. There are other algorithms aimed to optimize HEVC in different way, for example, Yan et al. [23] proposed a parallel framework to decide coding unit trees on many-core processors. Yan et al. [24] used a directed acyclic graph to parallelize CTUs to optimize HEVC intra-prediction.

A significant number of studies have focused on decreasing HEVC computation complexity by optimizing the intra-mode decision process. For example, Liu et al. [25] showed a fast mode decision algorithm that filters out unnecessary prediction units based on texture complexity and direction for HEVC intra-prediction, with texture complexity defined as the difference between the current pixel and its surrounding pixels, and the standard deviation of the current CU block, saving average time of 38% compared with HM 10.0. Ruiz et al. [26] presented a texture orientation detection algorithm by computing the dominant gradient and reducing the unnecessary directional candidate modes in the RDO process, saving average time of 30.1% compared with HM 14.0. Jiang et al. [27] calculated gradient directions and generated a gradient-mode histogram for each CU. The distribution of the histogram leaves only a small number of the candidate modes for the RMD and the RDO processes. Zhao et al. [28] took advantage of the direction information of the neighboring blocks to reduce the candidates in RDO process. Yan et al. [29] utilized early termination and pixel-based edge detection methods to reduce the number of candidates for the RDO process, saving average time of 23.52% compared with HM 7.0. Silva et al. [30] proposed an algorithm which took into account the edge direction information and explored the correlation of intra-modes across levels of the HEVC hierarchical tree structure. Chen et al. [31] proposed a candidate mode selection algorithm that adds kernel density estimation to the histogram calculation. Chen et al. [32] proposed a fast mode depth decision algorithm based on edge detection and reconfiguration to decrease the computation complexity in intra-prediction. In general, these papers [30–32] all considered edge detection to speed up the intra-prediction mode process. Motra et al. [33] planned to reduce 35 modes to 17 modes following the direction information of the co-located neighboring blocks of the previous frame along with neighboring blocks of the

current frame to speed up the intra-mode decision process, saving time of 23% compared with HM 6.0.

In addition to these fast intra-mode algorithms, several algorithms that combine early CU pruning and intra-mode decision together, as Shen et al. [34] did when they skipped a number of specific depth levels rarely used in spatially nearby CUs. RD cost and prediction mode correlations among different depth levels or spatially near CUs also exist. Their algorithm got on average time saving of 21.1 and 1.74% BD-BR increase compared with HM 5.0. Liao et al. [35] combined CU depth information with the order of most possible modes (MPMs) skipping some unlikely modes to save 31% encoder time compared with HM16.0. Tian et al. [36] utilized the cost of two candidate modes and the texture consistency of neighborhood and current PU to reduce modes in RMD and RDO process, achieving an average time reduction of 30% compared with HM16.0. However, although they decreased the number of modes included in the RMD process in [33, 35, 36], the method they adapted is different from ours. Unlike in previous studies, the current study takes advantage of RMD in two stages, decreasing mode selection complexity and using CU coding bits as CU partition judgment to decide whether a CU needs to subdivide further into smaller CUs. Comprehensively, our algorithm achieved better compression efficiency and easier implementation.

2 Proposed method and experimental setup

This study is aimed to optimize the HEVC, speed up encoding efficiency, and maintain the almost equal encoding quality simultaneously. In this study, we speed up intra-prediction in two phases. The first stage involves an adaptive mode decision algorithm while the second phase utilizes a fast CU early termination method. The goal of our algorithm is to perform the RMD process in two stages, namely (1) finding the most promising nodes in the first stage and (2) adding the modes adjacent to the most promising ones during the second stage of the RMD, with two or three modes selected during the two stages being adapted into the RDO process.

The material we tested is 7988 frames of all sequences (classes A to E) with different resolution. We first analyze the statistical data we collected, then find the appropriate thresholds in statistical way, last propose the fast algorithm and do experiments. We coded up to 7988 frames of all sequences (classes A to E) with the test conditions being "All Intra-Main" [37]. QP values are set to 22, 27, 32, and 37 for all frames. The tested computer is GreatWall with a Windows 10 operating system running on Inter Core i5 (TM) CPU of 3.2 GHz and 4 GB RAM. Coding performance is measured in terms of Bjontegaard delta metrics (BD-rate, BD-PSNR) and time reduction rate.

The remainder of this paper is organized as follows: Section 3 describes the whole proposed algorithm in detail. Section 4 illustrates the experiment results and discussion. Finally, the conclusions are presented in Section 5.

3 Proposed algorithm

This paper presents the adaptive mode decision and CU partition early termination algorithm in two parts. First,

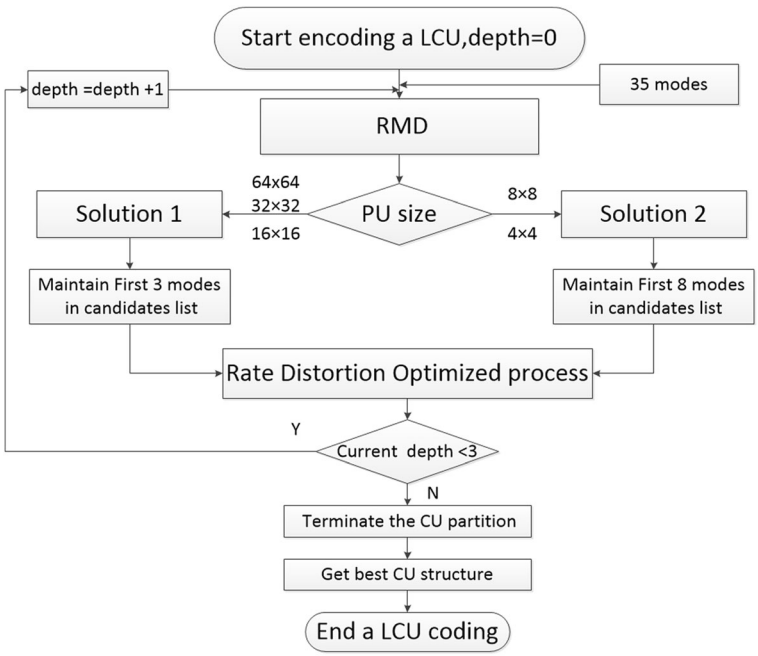

Fig. 4 Architecture of LCU coding of HM

the RMD process is divided into two stages according to the mathematical statistics results and early CU partition termination using thresholds based on the CU total coding bits. Figure 4 shows the procedure architecture of LCU coding of HM intra-prediction, and Fig. 5 illustrates the architecture of the proposed algorithm. Figure 5 clarifies the proposed algorithm more clearly compared with Fig. 4.

3.1 RMD operation

In the current HEVC test model, two steps are used to decide on the best intra-coding mode [38]. First, a subset of all intra-prediction modes is obtained by calculating the SATD in RMD process. The first phase is adapted to relieve the burden of the encoder in intra-coding. As shown in Table 1, the number (N) of the most promising modes subsets is pre-determined to be 8 for 4×4 and 8×8 PUs and 3 for 16×16, 32×32, and 64×64 PUs using the following RMD evaluation cost equation:

$$J_{\mathrm{RMD}} = \mathrm{SATD} + \lambda_{\mathrm{pred}} B_{\mathrm{pred}}, \tag{1}$$

where SATD is calculated by deriving the sum of the absolute Hadamard transform residual and B_{pred} is the number of bits needed to code the prediction mode information, and λ_{pred} presents the Lagrangian constant values variations with quantization parameters. The most possible modes (MPMs) derived from neighboring blocks are added to the subset. Second, in the best mode decision phase, the RD cost of each mode in the subset is computed to find the best mode, which can be calculated using the following formula:

$$J_{\mathrm{BMD}} = \mathrm{SSE} + \lambda_{\mathrm{mode}} \mathrm{B}_{\mathrm{mode}}, \tag{2}$$

where SSE is the sum of squared errors between the original input image block and the predicted block, B_{mode} is the number of bits needed for coding the current CU by the corresponding mode, and λ_{mode} is the Lagrange multiplier. The mode with the least RD cost is identified

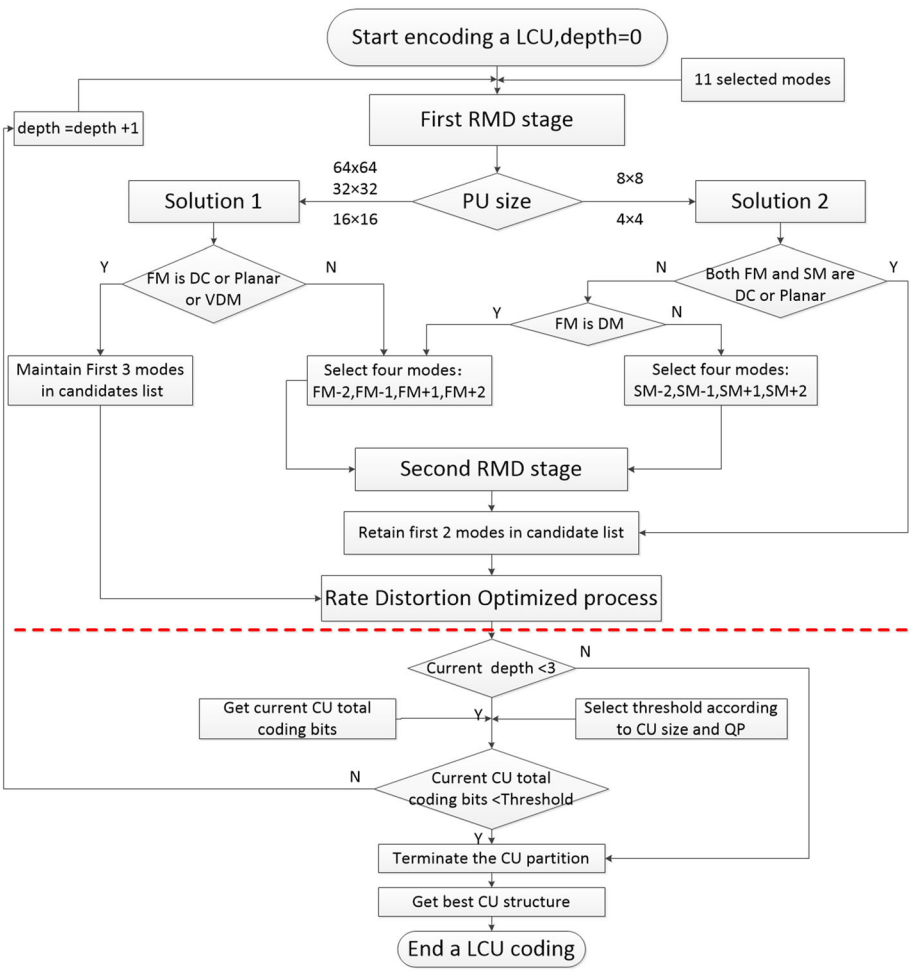

Fig. 5 Architecture of proposed algorithm. FM is the first mode in candidate list, SM is the second mode in candidate list, DM is the directional mode, and VDM is the vertical direction mode

Table 1 Most promising modes for different PU types

PU type	64 × 64	32 × 32	16 × 16	8 × 8	4 × 4
N	3	3	3	8	8

to be the best mode for finding the optimal residual quad-tree (RQT) structure. SATD and RDO are the key operations for selecting the best mode [39]. The SATD operation aim in part to reduce RDO complexity. However, the computational complexity increased dramatically because all possible combinations of the mode candidates are calculated to determine the optimal RD cost using the Lagrange multiplier. To decrease HEVC complexity and speed up the process in determining the best intra-mode, we collected a considerable amount of data and observed the relationship between the most promising mode and the first *N* modes in the candidate list yielded by the RMD process. In our algorithm, we tested 11 instead of 35 modes in the RMD process, with the 11 modes being a DC, a PLANAR, and nine directional modes, which were selected by equal intervals. These directional modes are labeled using red in Fig. 6.

3.2 Prepared work

We analyzed five typical sequences with different motion and texture information and collected the best CU modes, as well as 35 modes corresponding to every RMD evaluation cost in each depth, with the best CU mode being nearly identical to the one listed before it in the candidate list, where the modes are ranked from the one as the lowest RMD evaluation cost to the one with the highest. The statistics collected from five typical sequences in HM were shown in Table 2; looking at the statistical data, it appears that the best CU mode is almost the same as the first mode (FM) or the second mode (SM) in the candidate list, especially in cases where both the first and second candidate modes are DC or PLANAR modes.

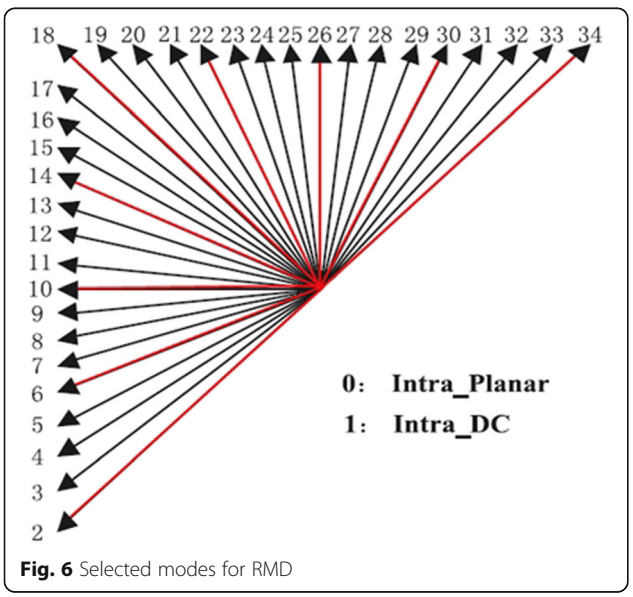

Fig. 6 Selected modes for RMD

To further illustrate the relationship between two or three modes in the candidate list from the RMD process and the optimal mode in the RDO process, the relationship among PUs of different sizes, 64 × 64, 32 × 32, and 16 × 16, have to be explained. The results show that if the FM is a DC, PLANAR, or vertical directional mode (VDM), one of the three selected modes by the RMD process is the final best mode. If the first candidate mode is a directional mode, the selected best mode by the RDO process is the FM or one of the FM's neighboring directional modes. In the case of 8 × 8 and 4 × 4 PUs, if the FM is a DC or PLANAR mode; the probability of the FM being selected as the best mode is almost 95%. If the first candidate mode is a directional mode, the selected best mode by the RDO process is most likely the FM or the neighboring directional modes of the FM. If the FM is a DC or a PLANAR mode and the SM is a directional mode, the selected best mode by the RDO process will most likely be the FM, SM, or any of the neighboring directional modes of the SM. We calculated the number of CUs that matched the aforementioned rules in every depth in the first frame of each of the five sequences to narrate the scheme more clearly and concisely. The CU matching ratio in every depth is displayed in Fig. 7, with the matching ratio α calculated as follows:

$$\alpha = \frac{\text{number of mathed CU}}{\text{number of total CU}} \times 100\%. \tag{3}$$

Figure 7 shows how the best CU modes obtain high matching ratios with the aforementioned rules, with all of the ratios being over 75% and fluctuating only minimally across different sequences. The texture of some PUs in natural video images changes slowly and smoothly, with DC and PLANAR modes being effective at intra-prediction when applied to a homogeneous area. In other areas, the texture of PUs changes in a particular direction; the rough angle of which can be found in the RMD process and the optical direction of which is decided by the RDO process. These observations led to the proposal of an adaptive mode

Table 2 The percentage (%) of the best mode is FM or SM

Class	Sequences	FM or SM	Both FM and SM are DC or planar
A	PeopleOnStreet	75	93
B	BasketballDrive	73	85
C	BasketballDrill	71	82
D	BlowingBubbles	68	79
E	FourPeople	74	78

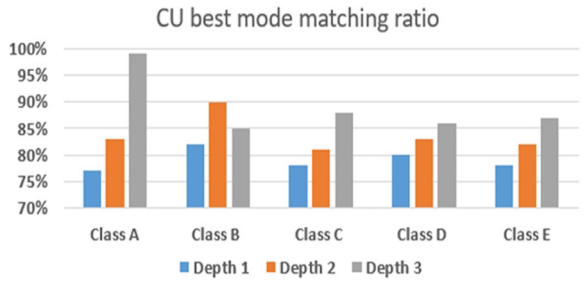

Fig. 7 The best CU mode matching ratio of different sequences and different depth

decision algorithm based on statistical data and image texture variation characteristics analysis.

3.3 Specific implementation of intra-mode decision algorithm

We divided the RMD process into two steps, following the image texture characteristics and statistical data. The specific details of the implementation are shown as the part of the flowchart above the red broken line in Fig. 5, details are further illustrated as following.

First, the 33 directional modes are reduced to 9 equally spaced modes, as marked by the red line in Fig. 6. These nine modes, a DC, and a PLANAR totally 11 modes will be tested using the RMD process to easily find the mode with the least J_{RMD}, namely the FM. Second, we will judge whether the FM is a DC or a PLANAR. If the FM is a DC or a PLANAR, this PU best mode is likely to adapt the form of a DC or a PLANAR. If the FM is a directional mode, we may speculate that this PU will most likely be a mode adjacent to the FM. We then add the modes adjacent to the FM to be tested during the RMD process, forming a new mode candidate list from which we select two or three modes during the RDO process to obtain the optimal mode. Computing for different PU types, we give two detailed solutions to acquire the optimal mode faster than HEVC.

Solution 1 is applied to PUs sized 64×64, 32×32, and 16×16 after the FM is obtained during the RMD process. If the FM is a DC, a PLANAR, or a VDM, we maintain three modes in the candidate list, with these three modes added by MPMs being tested during the RDO process. If the FM is a directional mode, we add four modes adjacent to the FM as FM − 1, FM − 2, FM + 1, and FM + 2 to be tested during the RMD process again. Then, the modes in the candidate list will be rearranged in an ascending order based on the value of their J_{RMD}. Lastly, we choose two modes from the top of the candidate list to be tested during the full RDO process. Hence, only two modes instead of three will be tested during the RDO process.

Solution 2 is applied to PUs sized 8×8 and 4×4. In HEVC, eight modes selected by the RMD process add MPMs to the RDO process. In our algorithm, we judge whether the FM and the SM are both DC or PLANAR. If the FM is a DC or a PLANAR, we select two modes instead of eight for the RDO process. If the SM is a directional mode, we add four modes adjacent to the SM as SM − 1, SM − 2, SM + 1, and SM + 2 to be tested during the RMD process again, and then, we choose two modes at the top of candidate list which is arranged in an ascending order based on the value of J_{RMD}. If the FM is a directional mode, we add four modes adjacent to the FM as FM − 1, FM − 2, FM + 2, and FM + 1 to be tested during the RMD process again. We then choose two modes at the top of candidate list which is arranged in an ascending order based on the value of J_{RMD}.

Supplementary explanation, the FM or SM is mode number 2 or 34, their neighboring modes only have two modes, respectively.

Table 3 shows the difference of the modes we selected is tested in RMD process and RDO process in our proposed algorithm compared with HM.

Combining the two solutions, significant encoding time is saved over HEVC while the encoder quality remains nearly the same. The pseudo-code is provided as following, corresponding the part of the flowchart above the red broken line in Fig. 5.

Intra-mode decision algorithm pseudo-code

```
First RMD stage: put DC, planar, and 9 modes labeled in Fig. 6 into first RMD stage to get candidate list
Second RMD stage:
Solution 1: if (PU size = = 64 × 64||32 × 32||16 × 16)
  If (FM = = DC||planar||VDM)
  Maintain first 3 modes in candidate list into RDO
  Else
  Select FM − 2, FM − 1, FM + 1, and FM + 2 modes into second RMD stage
  Get reconstructed candidate list
  Retain first 2 modes in candidate list into RDO
Solution 2: else
  If ((FM = = DC && SM = = planar)||(FM = = planar && SM = = DC))
  Maintain first 2 modes in candidate list into RDO
  Else
  If (FM = = DM)
  Select FM − 2, FM − 1, FM + 1, and FM + 2 modes into second RMD stage
  Get reconstructed candidate list
  Retain first 2 modes in candidate list into RDO
  Else
  Select SM − 2, SM − 1, SM + 1, and SM + 2 modes into second RMD stage
  Get reconstructed candidate list
  Retain first 2 modes in candidate list into RDO
```

3.4 Fast CU partition early termination algorithm

To save more time, the CU partition rule is studied further. As shown in Fig. 3, every frame in the picture will be divided into different sized CUs, the best

Table 3 The difference of modes will be tested in RMD and RDO process in HM and proposed

	RMD	RDO	
		PU size: 64 × 64, 32 × 32, and 16 × 16	PU size: 8 × 8 and 4 × 4
HM	35	3 + MPMs	8 + MPMs
Proposed	11/11 + 4	3 + MPMs/2 + MPMs	2 + MPMs

combinations of which follow some rules. The CUs homogeneous in area and contain less complex information require lesser coding bits. CUs usually heterogeneous in area and contain more dynamic information require more coding bits as shown in Fig. 8a. The first LCU in Fig. 8a is sized 64 × 64, meaning its best depth is 0 and its coding bits count is 95. The second LCU is divided into different sized CU combinations, with the largest depth at 3. The coding bits of every CU are shown in Fig. 8b. The first LCU is an obviously homogeneous area with less dynamic information that requires less coding bits. The second LCU is a heterogeneous area that contains more dynamic information and requires more coding bits. The heterogeneous LCU with more coding bits is usually divided into sub-CUs, with the heterogeneous sub-CUs with more coding bits being subdivided further into smaller sub-CUs until the optimal combination is achieved. This optimal combination is the least resource-consuming and requires the least amount of data transfer. Every CU requires different coding bits, and we have simplified the process of finding the best intra-prediction mode to obtain the CU coding bits.

If the CUs in each level are used in the process of finding the best combination, a considerable amount of time is wasted. Hence, we propose a CU early pruning algorithm that takes advantage of obtaining coding bits earlier to speed up the process of finding the best CU combination. HEVC uses essentially the same uniform reconstruction quantization scheme controlled by a quantization parameter (QP) as in H.264/MPEG-4 AVC. The range of QP values is defined from 0 to 51, and an increase by 6 doubles the quantization step size such that the mapping of QP values to step sizes is approximately logarithmic. The QP values affect the number of CU's total coding bits required, if the QP is larger, the CU's total coding bits is smaller. The QP values in this paper are defined as 22, 27, 32, and 37 recommended by [37] to collect statistical data and test all sequences. We count the demand coding bits of different CUs sized 64 × 64, 32 × 32, and 16 × 16 under different quantization parameters (QP) derived through comparative analysis. By setting a certain coding bit threshold, we can obtain the best CU combinations more quickly than in HEVC.

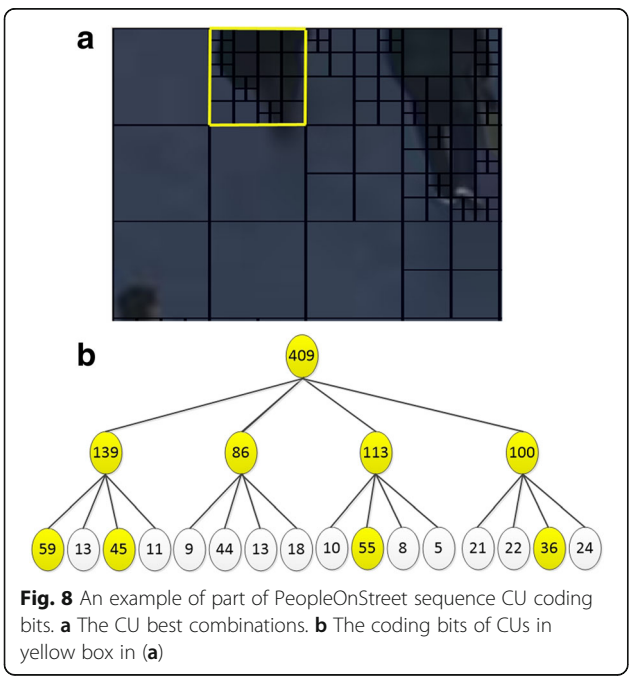

Fig. 8 An example of part of PeopleOnStreet sequence CU coding bits. **a** The CU best combinations. **b** The coding bits of CUs in yellow box in (**a**)

Part (b) of Fig. 8 shows the coding bits of part (a) of Fig. 8, where the LCU in the yellow box shows the CUs of every level with sizes 64 × 64, 32 × 32, and 16 × 16 from top to bottom, with the number of coding bits shown. All demarcated CUs are divided into four sub-CUs. The CUs outside the yellow mark are the best CU partitions.

We analyzed the demanding coding bits of different sized CUs using different sequences to obtain an accurate threshold and achieve higher compression efficiency and lower quality loss. The thresholds of the CUs sized 64 × 64, 32 × 32, and 16 × 16 under different QPs are shown in Table 4. We analyzed the accuracy in each threshold and described the results in Fig. 9. The error of our algorithm exists that the CUs' coding bits is smaller than the thresholds we set; these CUs are not divided into four next-level CUs in our algorithm. Instead, these CUs are divided into four next-level CUs in HEVC, as shown in Fig. 8 (32 × 32 CU 139 coding bits).

Table 4 Thresholds of the CUs with different sizes and different QPs

QP	22	27	32	37
CU size				
64 × 64	850	500	200	100
32 × 32	400	200	100	50
16 × 16	120	80	45	35

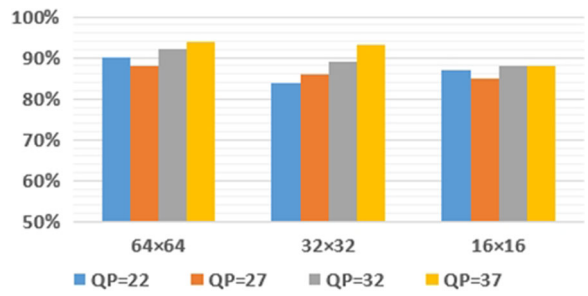

Fig. 9 Accuracy of early CU termination algorithm

The number of CU coding bits required under the best intra-mode can be obtained, and then, we judge whether the number of coding bits is smaller than the threshold set by our statistical data. If the number of coding bits is smaller than the threshold shown in Table 4, we can end the CU partition early. Otherwise, CU continues to be divided into four sub-CUs, as shown in the detailed implementation process illustrated under the red broken line in Fig. 5. The HEVC encoder achieves higher encoding efficiency by judging the amount of CU coding bits in advance.

4 Results and discussion

We use our proposed fast intra-mode decision and CU partition early termination algorithm on HM16.12 to evaluate the effectiveness of the algorithm. Time reduction is calculated by the following equation:

$$\Delta T = \frac{\mathrm{T}_{\mathrm{HM16.12}} - T_{\mathrm{proposed}}}{T_{\mathrm{HM16.12}}} \times 100\%, \tag{4}$$

where $T_{\mathrm{HM16.12}}$ is the coding time of HM16.12, T_{proposed} is the coding time of HM16.12 using the proposed algorithm, and $\Delta T$is the time reduction. The decrease in PSNR-Y is calculated using Eq. (5):

$$\Delta \mathrm{PSNR_Y} = \mathrm{PSNR_{HM16.12}} - \mathrm{PSNR_{proposed}}. \tag{5}$$

7988 frames of all sequences (classes A to E) with different resolution are tested to checkout our proposed algorithm. Table 5 shows that on average [40], the proposed algorithm achieved 53% time reduction, 1.7% BD-rate increase, and 0.08 dB BD-PSNR decrease. Table 5 also shows the experimental results of [35, 36], under same test condition, [35] achieved 31% time reduction on average, 0.7% BD-rate increase compared with HM 16.0, although [36] got 30% time reduction on average, and it brought 1.8% BD-rate increase compared with HM 16.0. The comparison data certified our proposed algorithm is efficient sufficiently. Generally, the different sequences obtain different time-saving percentages mainly because the different sequence frames have different detail changes and complexity. Table 5 shows

Table 5 Experiment results of our proposed algorithm and reference [35, 36]

		[35]		[36]		Proposed		
Class	Sequence	BD-rate _*Y* (%)	▵T (%)	BD-rate _*Y* (%)	▵T (%)	BD-rate _*Y* (%)	▵PSNR _*Y*	▵T (%)
A (4K)	PeopleOnStreet	0.7	32	2.2	33	1.4	0.08	50
	Traffic	0.7	31	2.0	31	1.5	0.08	52
B (1080P)	BasketballDrive	0.6	31	2.1	28	1.8	0.05	66
	BQTerrace	0.4	31	–	–	2.0	0.07	50
	Cactus	0.7	31	1.9	29	1.9	0.07	52
	Kimono	0.4	32	2.2	30	1.2	0.03	68
	ParkScene	0.5	31	2.2	33	0.9	0.07	50
C (WVGA)	BasketballDrill	0.6	30	1.4	30	1.9	0.08	50
	BQMall	0.8	31	1.9	33	1.6	0.09	47
	PartyScene	0.9	30	1.7	29	1.3	0.13	37
	RaceHorses	0.5	30	1.3	30	1.0	0.08	47
D (WQVGA)	BasketballPass	0.8	30	1.8	31	1.6	0.08	51
	BlowingBubbles	1.0	31	1.9	30	1.4	0.11	38
	BQSquare	1.0	31	1.7	29	1.7	0.15	39
	RaceHorses	0.8	30	1.7	28	1.4	0.1	40
E (720P)	Johnny	0.8	33	2.1	28	2.7	0.07	67
	KristenAndSara	0.8	31	2.2	29	2.7	0.09	66
	FourPeople	0.7	32	2.2	28	2.8	0.11	58
Average		0.7	31	1.8	30	1.7	0.08	53

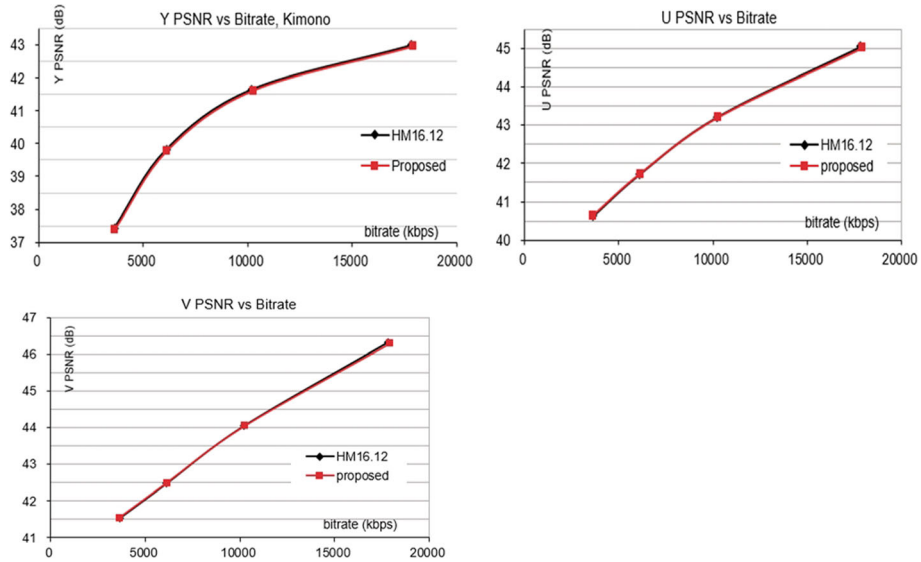

Fig. 10 RD-curves of "Kimono" under different QPs (22, 27, 32, 37)

our proposed algorithm reduces coding time of "Kimono" by up to 68%. Figure 10 shows that the RD curves of "Kimono" are almost the same as that of the original encoder. Our algorithm performs very well in sequences class A to D, especially in high-resolution sequences. Time reduction and quality rates are almost perfect, making the algorithm applicable for compressing videos that require real-time and high-resolution compression. However, the BD-rate of three sequences under class E is not the same or better than the others. We also explore why the sequences of PartyScene, BlowingBubbles, and BQSquare have time reduction rates that fall below 40%. Compared with other sequences with higher time reduction, we find that the CUs of these sequences will most likely be divided into smaller CUs, indicating that their corresponding depths are larger than that of other sequences, and hence, the threshold set to end the CU partition earlier does not play an important role in the compression process. Overall, the proposed algorithm significantly improved coding efficiency while maintaining the average RD performance at nearly the same level.

5 Conclusions

This paper presents an adaptive mode decision and CU partition early termination algorithm for alleviating the computational complexity of the HEVC intra-encoder. The proposed algorithm is performed on HEVC reference software HM16.12. The adaptive mode decision and CU partition early termination algorithms both reduce encoder time in different HEVC processes. The aforementioned experiment results show that through the implementation of the RMD process as a two-stage process, decreasing the number of promising modes during the RDO process, and using the coding bits generated during the HEVC coding process, our proposed algorithm can improve intra-coding efficiency. Our algorithm reduced total coding time by 53% while maintaining coding performance at nearly the same level as that of the original HEVC encoder.

Abbreviations

BD-Rate: Bjontegaard delta rate; CABAC: Context-adaptive binary arithmetic coding; CTU: Coding tree unit; CU: Coding unit; FM: First mode; HEVC: High Efficiency Video Coding; HM: HEVC test model; JCT-VC: Joint Collaborative Team on Video Coding; LCU: Largest coding unit; MB: Macroblock; MPEG: Moving Picture Exports Group; MPMs: Most possible modes; PSNR: Peak to signal noise ratio; PU: Prediction unit; QP: Quantization parameters; RD: Rate distortion; RDO: Rate-distortion operation; RMD: Rough Mode Decision; RQT: Residual quad-tree; SAO: Sample adaptive offsets; SATD: Sum of the absolute transformed difference; SM: Second mode; SVM: Support vector machine; TU: Transform unit; VCEG: Video Coding Experts Group; VDM: Vertical directional mode

Acknowledgements

Not applicable.

Availability of data and materials

The conclusion and comparison data of this article are included within the article.

Authors' contributions

MZ proposed the framework of this work, and XZ carried out the whole experiments and drafted the manuscript. ZL offered useful suggestions and helped to modify the manuscript. All authors read and approved the final manuscript.

Funding

This work is supported by the National Natural Science Foundation of China (no. 61370111), Beijing Municipal Natural Science Foundation (no. 4172020), Beijing Nova Programme (Z141101001814032), Beijing Youth Talent Project (CIT&TCD 201504001), and Beijing Municipal Education Commission General Program (KM201610009003).

Authors' information
MMZ: Doctor of Engineering, Professor, Master Instructor, Master of Communication and Information Systems. His major research interests include the Video codec, Embedded systems, Image processing, and Pattern recognition. He has authored or co-authored more than 30 refereed technical papers in international journals and conferences in the field of video coding, image processing, and pattern recognition. He holds 16 national patents and 2 monographs in the areas of image/video coding and communications.
XJZ: Studying master of North China University of Technology. Her major research is HEVC.
ZL: Doctor of Engineering, Master Instructor. His major research interests include the video codec, pattern recognition, and self-organizing network.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 16 May 2017 Accepted: 29 November 2017
Published online: 15 December 2017

References

1. GJ Sullivan, J-R Ohm, W-J Han, T Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
2. T Wiegand, GJ Sullivan, The H.264/AVC video coding standard. IEEE Signal Process. Mag. **24**(2), 148–153 (2007)
3. J Lainema, F Bossen, W-J Han, J Min, K Ugur, Intra coding of the HEVC standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1792–1801 (2012)
4. GJ Sullivan, J-R Ohm, HEVC software guidelines(Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, document JCTVC-H1001, 8th Meeting, San José, CA, USA), 2012
5. L Shen, Z Zhao, P An, Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Consum. Electron. Soc. **59**(1), 207–213 (2013)
6. MKJ Kim, *JCTVC-C067 TE9: report on large block structure testing (In proceedings of Third meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Guangzhou, China)* (2010), pp. 2–4
7. C Yan, H Xie, D Yang, J Yin, Y Zhang, Q Dai, Supervised hash coding with deep neural network for environment perception of intelligent vehicles. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–12 (2017)
8. H Bai, C Zhu, Y Zhao, Optimized multiple description lattice vector quantization for wavelet image coding. IEEE Trans. Circuits Syst. Video Technol. **17**(7), 912–917 (2007)
9. C Yan, H Xie, S Liu, J Yin, Y Zhang, Q Dai, Effective Uyghur language text detection in complex background images for traffic prompt identification. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–10 (2017)
10. H Bai, W Lin, M Zhang, A Wang, Y Zhao, Multiple description video coding based on human visual system characteristics. IEEE Trans. Circuits Syst. Video Technol. **24**(8), 1390–1394 (2014)
11. Y-J Ahn, T-J Hwang, D-G Sim, W-J Han, Implementation of fast HEVC encoder based on SIMD and data-level parallelism. EURASIP J. Image Video Process. **2014**, 16 (2014)
12. KY Min, W Lim, J Nam, D Sim, IV Bajić, Distributed video coding supporting hierarchical GOP structures with transmitted motion vectors. EURASIP J. Image Video Process. **2015**, 12 (2015)
13. C Yan, Y Zhang, J Xu, F Dai, J Zhang, Q Dai, F Wu, Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Trans. Circuits Syst. Video Technol. **24**(12), 2077–2089 (2014)
14. C Yan, Y Zhang, F Dai, X Wang, L Li, Q Dai, Parallel deblocking filter for HEVC on many-core processor. Electron. Lett. **50**(5), 367–368 (2014)
15. Z Peng, H Han, F Chen, G Jiang, M Yu, Joint processing and fast encoding algorithm for multi-view depth video. EURASIP J. Image Video Process. **2016**, 24 (2016)
16. T Nishikori, T Nakamura, T Yoshitome, K Mishiba, *A fast CU decision using image variance in HEVC intra coding (IEEE Symposium on Industrial Electronics & Applications (ISIEA))* (2013), pp. 52–56
17. C Bai, C Yuan, *Fast coding tree unit decision for HEVC intra coding (IEEE International Conference on Consumer Electronics, China)* (2013), pp. 28–31
18. SG Blasi, I Zupancic, E Izquierdo, E Peixoto, *Fast HEVC coding using reverse CU visiting (Picture Coding Symposium(PCS))* (2015), pp. 50–54
19. F Belghith, H Kibeya, MAB Ayed, N Masmoudi, Fast coding unit partitioning method based on edge detection for HEVC intra-coding. SIViP **10**(5), 811–818 (2016)
20. K Goswami, BG Kim, D Jun, SH Jung, SC Jin, Early coding unit-splitting termination algorithm for high efficiency video coding (HEVC). ETRI J. **36**(3), 407–417 (2014)
21. X Shen, L Yu, CU splitting early termination based on weighted SVM. EURASIP J. Image Video Process. **2013**, 4 (2013)
22. YF Cen, WL Wang, XW Yao, A fast CU depth decision mechanism for HEVC. Inf. Process. Lett. **115**(9), 719–724 (2015)
23. C Yan, Y Zhang, J Xu, F Dai, L Li, J Zhang, Q Dai, F Wu, A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Process Lett. **21**(5), 573–576 (2014)
24. C Yan, Y Zhang, F Dai, J Zhang, L Li, Q Dai, Efficient parallel HEVC intra prediction on many-core processor. Electron. Lett. **50**(11), 805–806 (2014)
25. Y Liu, X Liu, P Wang, *A texture complexity based fast prediction unit size selection algorithm for HEVC intra-coding (IEEE 17th International Conference on Computational Science and Engineering)* (2014), pp. 1585–1588
26. D Ruiz, G Fernández-Escribano, JL Martínez, P Cuenca, Fast intra mode decision algorithm based on texture orientation detection in HEVC. Signal Process. Image Commun. **44**(C), 12–28 (2016)
27. W Jiang, H Ma, Y Chen, *Gradient based fast mode decision algorithm for intra prediction in HEVC (IEEE, 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet))* (2012), pp. 1836–1840
28. L Zhao, L Zhang, S Ma, D Zhao, *Fast mode decision algorithm for intra prediction in HEVC (Visual Communications and Image Processing (VCIP))* (2011), pp. 1–4
29. S Yan, L Hong, W He, Q Wang, *Group-based fast mode decision algorithm for intra prediction in HEVC (Eighth International Conference on Signal Image Technology and Internet Based Systems)* (2012), pp. 225–229
30. TLD Silva, LADS Cruz, LV Agostini, *HEVC intra mode decision acceleration based on tree depth levels relationship (IEEE, Picture Coding Symposium(PCS))* (2013), pp. 277–280
31. G Chen, Z Liu, T Ikenaga, D Wang, *Fast HEVC intra mode decision using matching edge detector and Kernel density estimation alike histogram generation (IEEE International Symposium on Circuits and Systems(ISCAS))* (2013), pp. 53–56
32. G Chen, L Sun, Z Liu, T Ikenaga, *Fast mode and depth decision HEVC intra prediction based on edge detection and partitioning reconfiguration (International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS))* (2013), pp. 38–41
33. A-S Motra, A Gupta, M Shukla, P Bansal, V Bansal, *Fast intra mode decision for HEVC video encoder, (SoftCOM 20th International Conference on Software, Telecommunications and Computer Networks)* (2012), pp. 1–5
34. L Shen, Z Zhang, P An, Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Trans. Consum. Electron. **59**(1), 207–213 (2013)
35. W Liao, D Yang, Z Chen, *A fast mode decision algorithm for HEVC intra prediction, (IEEE Visual Communications and Image Processing (VCIP))* (2016), pp. 1–4
36. R Tian, Y Zhang, R Fan, G Wang, *Adaptive fast mode decision for HEVC intra coding, (Digital Image Computing Techniques and Applications (DICTA))* (2016), pp. 1–6
37. F Bossen, *Common test conditions and software reference configurations, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JCT1/SC29/WG11, (Doc. JCTVC-B300, 2nd Meeting, Geneva, CH, 21-28 July)* (2010), pp. 14–23
38. M Zhang, J Qu, H Bai, Entropy-based fast largest coding unit partition algorithm in high-efficiency video coding. Entropy **15**(6), 2277–2287 (2013)
39. M Zhang, C Zhao, J-Z Xu, *An adaptive fast intra mode decision in HEVC, (IEEE International Conference on Image Processing (ICIP))* (2012), pp. 221–224
40. G Bjontegaard, *Calculation of average PSNR differences between RD-curves (doc.VCEG-M33, in ITU-T VCEG 13th Meeting, Austin)* (2001), pp. 2–4

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com