

RESEARCH

Open Access



A fast HEVC intra-coding algorithm based on texture homogeneity and spatio-temporal correlation

Xin Lu^{1*}, Chang Yu¹ and Xuesong Jin²

Abstract

The high efficiency video coding (HEVC) standard supports a flexible coding tree unit (CTU) partitioning structure, and thus coding efficiency is improved significantly. However, the use of this technique inevitably results in greatly increased encoding complexity. In order to reduce the complexity of intra-coding, we propose a hybrid scheme consisting of fast coding unit (CU) size decision and fast prediction unit (PU) mode decision processes. An adaptive method is utilised to measure the homogeneity of video content thus avoiding unnecessary rate distortion (RD) evaluations. The depth range to be tested is narrowed based on the partitioning parameters of the spatially adjacent CUs and the temporally co-located CU. Furthermore, the mode correlation between neighbouring frames and between adjacent coding levels in the same quad-tree structure is taken into account to predict the most probable directional mode. The number of candidate PU modes is further decreased according to the Hadamard cost. Experimental results illustrate that our scheme achieves a significant reduction in computational complexity of HEVC intra-coding. Compared with the HM encoder, the encoding time is reduced by up to 71% with negligible degradation in coding efficiency.

Keywords: HEVC, CU size decision, Prediction mode decision, Fast implementation

1 Introduction

With the increased demand for capturing and displaying high- and ultra-high-definition videos, video coding standards with high compression efficiency are required to accommodate the increased video resolution and frame rate.

HEVC was developed by the Joint Collaborative Team on Video Coding (JCT-VC) [1] and follows the conventional block-based hybrid video coding framework as adopted in previous standards. In order to achieve good coding performance, a number of advanced coding tools are incorporated, including a flexible quad-tree coding block structure, and up to 35 intra-prediction directions for each PU. HEVC employs several new coding structures, including CU, PU and transform unit (TU). A CU can be coded with various sizes, from the largest CU (LCU) size of 64×64 to the smallest CU (SCU) size of

8×8 . Each CU can be recursively split into four smaller blocks. A CU can be further partitioned into PUs and TUs, and the partitioning is implemented in a recursive manner within a quad-tree hierarchy. In order to determine the best coding mode for each PU, all candidate modes are exhaustively examined by calculating the RD cost. These techniques lead to dramatically increased encoding complexity, and this makes the implementation of HEVC in real-time applications difficult.

As the number of candidate modes is increased significantly compared to previous standards, the CU size decision and the PU mode decision account for most of the overall encoding time. Therefore, when determining the optimal prediction mode for a CU, an effective fast method to reduce the number of candidates in the rate distortion optimisation (RDO) process is desirable. In this study, our objective is to develop a computational complexity reduction scheme specifically designed for HEVC intra-coding and provide efficient solutions for the CU size and coding mode decisions. The proposed scheme utilises the texture complexity of the current

*Correspondence: xin@hit.edu.cn

¹School of Electronics and Information Engineering, Harbin Institute of Technology, No. 92 West Dazhi Street, Nangang District, Harbin, China, 150001
Full list of author information is available at the end of the article

LCU, the partitioning information of spatially neighbouring LCUs and the co-located CU in the previous frame to gradually narrow the CU depth range. The PU coding mode dependencies between coding depth levels and between neighbouring frames are exploited to eliminate the unlikely candidate modes. In addition, the Hadamard cost is employed to reduce the mode candidates that need to be evaluated in the residual-quad-tree (RQT) process. Simulation results indicate that the proposed algorithm can greatly reduce the computational complexity without any significant penalty in coding efficiency.

The rest of this paper is organised as follows. Related work in the field of HEVC fast implementation is discussed in the next section. Section 3 briefly introduces the intra-coding techniques in HEVC, and the proposed fast CU depth decision method is presented in Section 4. Section 5 describes the proposed fast PU mode decision algorithm, and Section 6 provides the structure of the overall fast algorithm. The simulation configuration is presented in Section 7. The extensive experimental results are given in Section 8, and the conclusions are drawn in Section 9.

2 Related work

Many algorithms have been proposed to simplify the intra-coding process in HEVC. These are categorised into mainly three types: fast coding depth decision [2–8], fast prediction mode decision [9–19] and hybrid fast scheme [20–26]. Some typical algorithms in each category are described as follows:

- **Fast coding depth decision:** In [2], Min et al. suggested an early termination strategy for CU partitioning, in which the local and global edge complexities were exploited. Shen et al. [3] proposed a fast CU size decision scheme, where large CU sizes can be bypassed depending on the texture homogeneity. In [4], a fast CU depth prediction method was proposed for HEVC screen content compression. The temporal correlation of co-located CUs was exploited to predict the most likely mode, and an adaptive search step approach was used to further accelerate the block matching process of intra block copy (IBC) mode.
- **Fast prediction mode decision:** The fast mode decision proposed by Gao [9] employed the spatial correlation between neighbouring blocks. In [10], a computational complexity reduction scheme for intra-coding was proposed by Jamali, where the edge property, spatial correlation between neighbouring PUs, and the classification of SATD costs were used. In [11], the edge direction information was exploited to define a reduced set of coding modes from which the best prediction mode is finally determined. In

[12], a scheme was suggested to reduce the encoding time for lossless intra-coding in HEVC. Sanchez et al. employed differential pulse coding to reduce the number of modes to be evaluated.

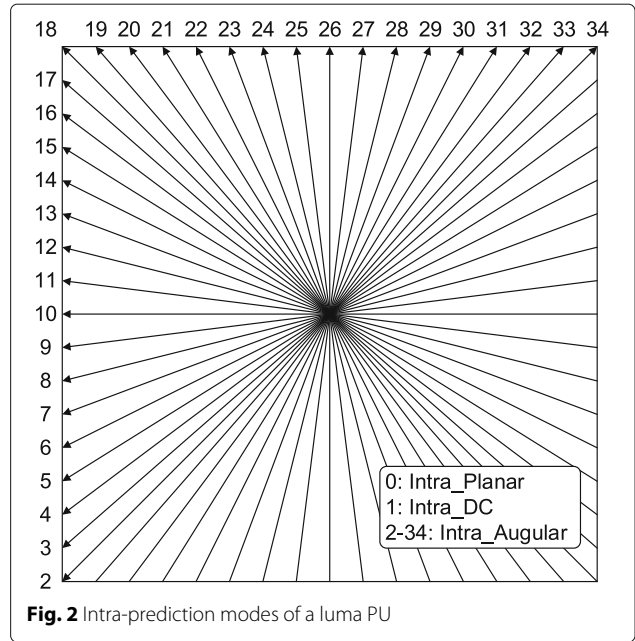
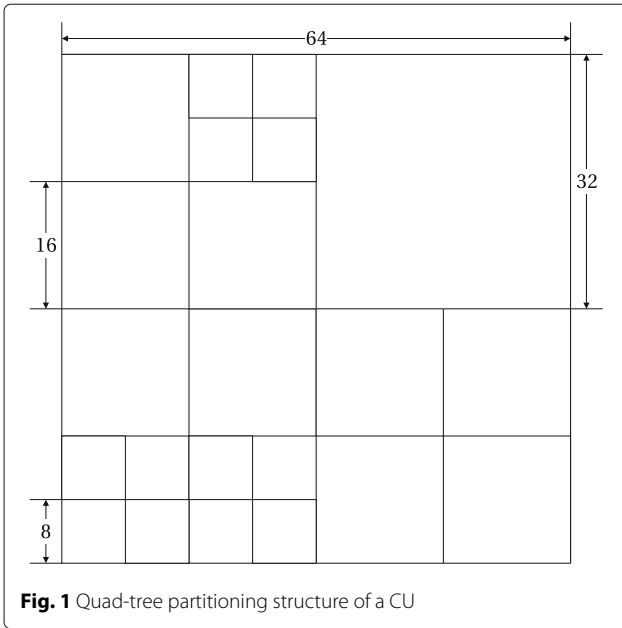
- **Hybrid fast scheme:** To accelerate the intra-coding process, the depth information of neighbouring CUs and the mode correlation between coding layers were employed in Shang's work [20]. In [21], Zhang et al. suggested a fast intra-mode and CU size decision algorithm for HEVC, where a gradient-based method was proposed for the intra-mode decision, and the fast CU size decision was achieved by two linear support vector machines (SVMs). The algorithm proposed by Zhao [22] consisted of early termination of CU depth, and fast PU intra-mode decision and TU depth restriction, where the correlation of prediction modes between spatially neighbouring PUs was employed to speed up the PU mode decision. Lei et al. proposed a fast intra-prediction method for HEVC-based screen content coding in [23]. The CUs were first classified into two types, for each of which, different strategies were implemented to eliminate inappropriate candidate modes.

Although many approaches have been proposed for HEVC intra-coding, the inter-level and spatio-temporal correlations, as well as the texture information, have not been fully exploited. In addition, most existing algorithms only provide improvements to the encoding method after the rough mode decision (RMD) process. If the above two issues can be addressed, it is anticipated that the computational requirements can be reduced further. Therefore, we propose a hybrid scheme consisting of fast CU size decision and fast PU mode decision. The scheme avoids the evaluation of unlikely partition sizes and prediction modes and skips the exhaustive examination at different coding depths. The main contributions can be summarised as follows: (1) The temporal correlation and the correlation between coding depth levels are jointly considered to reduce the computational requirement; (2) the energy distribution property and the Hadamard cost are exploited to eliminate unlikely coding depth levels and candidate modes; (3) the coding processes both prior to RMD and after RMD are improved.

3 A brief overview of HEVC intra-coding decision

3.1 CU partition

HEVC adopts a highly flexible CU partitioning structure, which is similar to the macroblock (MB) structure in H.264/AVC. Coding units can be of various sizes, namely, 64×64 , 32×32 , 16×16 and 8×8 , which correspond to coding depth levels 0, 1, 2 and 3, respectively [27]. Figure 1 presents the architecture of the quad-tree structure.



When encoding a video sequence, the first step is to segment each frame into non-overlapping LCUs. A LCU of size 64×64 can be split into four smaller CUs of size 32×32 , each of which can be further divided recursively. The partition of a CU is terminated when the minimum CU size is reached. This quad-tree structure enables efficient intra-prediction for regions with widely different texture.

3.2 PU prediction mode

Five sizes of PU are supported in HEVC intra-coding: 64×64 , 32×32 , 16×16 , 8×8 and 4×4 . In order to accommodate various textural features and obtain an accurate intra-prediction, HEVC offers up to 35 luma prediction modes, including DC, planar and 33 angular directional modes, as shown in Fig. 2. Planar mode is suitable for smooth regions with slow changes, while the DC mode is beneficial for areas representing homogeneous texture.

3.3 Intra-coding decision process

In HEVC, the best coding size and mode are determined by the following steps:

(1) Rough mode decision (RMD)

All 35 intra-prediction modes are investigated, based on the Hadamard cost, to construct the initial candidate list. This process can be described by Eq. (1):

$$\begin{aligned}
 HCOST_{MODE} &= SATD_{MODE}(\omega_k, \tilde{\omega}_k) + \lambda \cdot \tilde{R}_{MODE} \\
 SATD_{MODE}(\omega_k, \tilde{\omega}_k) &= \left(\sum |\mathbf{H}(\omega_k, \tilde{\omega}_k)| \right) / 2
 \end{aligned}
 \tag{1}$$

where $HCOST_{MODE}$ represents the Hadamard cost; $SATD_{MODE}$ indicates the sum of absolute Hadamard transformed differences (SATD); \tilde{R}_{MODE} is the estimated bit consumption; λ denotes the Lagrangian multiplier; ω_k represents an original PU at time k and $\tilde{\omega}_k$ is the corresponding reconstructed PU; and \mathbf{H} is the Hadamard transform.

(2) Combine with most probable mode (MPM)

The set of MPMs consists of the coding modes of adjacent PUs located to the left and above of the current PU. If these modes have yet to be included in the current candidate list, they are inserted.

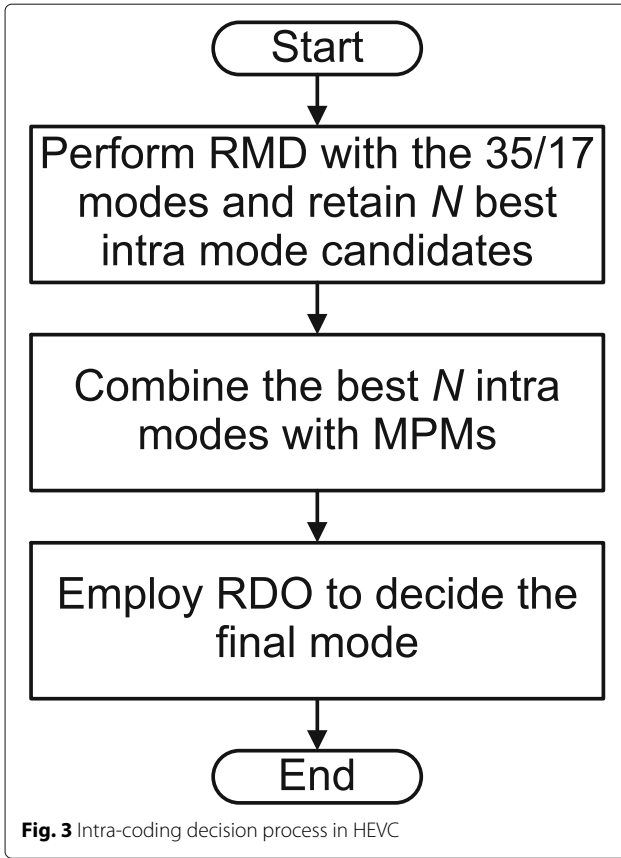
(3) RDO-based intra-coding decision

All the prediction modes in the candidate list are evaluated in the RDO process. Finally, the mode which produces the minimum RD cost is selected as the best coding mode.

The flowchart in Fig. 3 illustrates the process of intra-mode decision in HEVC. It can be seen that, when performing the RMD in the intra-mode decision process, the Hadamard costs of all 35/17 prediction modes for each PU are calculated. This results in a large computational requirement, and the encoding time is thus significantly increased. Therefore, in order to reduce the complexity of HEVC encoders, efficient intra-mode decision algorithms become even more important.

4 Fast CU size decision

In the process of HEVC intra-prediction coding, the encoder recursively traverses all CU depths and the RD costs of CUs at each coding depth level are evaluated to determine the best CU size. In a given LCU of size 64×64 ,



the number of CUs of sizes 64×64 , 32×32 , 16×16 and 8×8 that require the calculation of the RD cost is 1, 4, 16 and 64, respectively. This is about 2.5 times more than that in H.264/AVC. It can be seen that a total number of 85 RD cost calculations is required for each LCU. This is a very time-consuming process and becomes a significant part of the computational complexity.

In practice, a number of unlikely CU sizes can be eliminated in advance, and it is then unnecessary to calculate the RD costs for CUs of all possible partitioning sizes. In order to reduce the computational requirements and to speed up the intra-coding process, an efficient termination and bypass strategy is desirable.

4.1 Energy distribution and textural complexity

The spatial homogeneity of a CU can be characterised by its energy distribution in the frequency domain. Therefore, we utilise the alternating current (AC) coefficients energy to evaluate the textural complexity of a LCU. For a homogeneous area of an image, the low-frequency components tend to contain the majority of the frequency domain energy, whereas for a region comprising rich detail, more DCT energy is distributed over other AC coefficients.

According to the principle of energy conservation, the AC coefficients energy (E_{AC}) of a $sl \times sl$ CU is mathematically expressed as

$$E_{AC} = \sum_{x=0}^{sl-1} \sum_{y=0}^{sl-1} f^2(x, y) - \frac{1}{sl^2} \left(\sum_{x=0}^{sl-1} \sum_{y=0}^{sl-1} f(x, y) \right)^2 \quad (2)$$

where $f(x, y)$ denotes the intensity of a pixel sample at position (x, y) .

It should be noted that the maximum possible value, E_{max} , of a CU's AC energy can be determined in advance. E_{max} is obtained from the CU comprising a checkerboard pattern in which the intensity of every adjacent pixel sample is the allowable minimum (f_{min}) and maximum (f_{max}) value alternately. Therefore, the formulation for calculating E_{max} of a $sl \times sl$ CU is

$$E_{max} = \frac{sl^2}{2} \left[(f_{max}^2 + f_{min}^2) - \frac{1}{2} (f_{max} + f_{min})^2 \right] \quad (3)$$

Consequently, we define R_{CU} as the criterion to assess the textural complexity of a CU, which is shown in Eq. (4).

$$R_{CU} = \frac{\ln(E_{AC})}{\ln(E_{max})} \quad (4)$$

In Eq. (4), both E_{AC} and E_{max} are linearised by the natural logarithm such that the range of R_{CU} can be uniformly distributed.

The AC energy can be employed to measure the textural complexity of a LCU. Generally, a large R_{CU} value indicates that the LCU contains high detail, and smaller CU sizes at upper coding depth levels are more appropriate for these areas. On the contrary, homogeneous regions tend to generate smaller R_{CU} values, and lower coding depth levels are more beneficial.

4.2 Adaptive dual-threshold scheme

In order to achieve early termination by skipping unlikely coding depth levels, an adaptive dual-threshold scheme is suggested. In the proposed algorithm, we employ two AC energy thresholds to determine the spatial homogeneity of a LCU.

When the value of R_{CU} exceeds an upper threshold T_U , rich spatial detail is represented in the LCU, and the evaluation of large CU sizes can be omitted. When R_{CU} is below a lower threshold T_L , the LCU tends to contain low texture detail, and higher coding depth levels can be eliminated in advance.

It is required to select the appropriate thresholds for AC energy, such that the number of RD evaluations can be reduced and the encoding time minimised, while the compression efficiency is not adversely affected. The proposed dual-threshold scheme allows the two thresholds to be adaptively adjusted and updated, thus achieving a

balance between coding performance and computational complexity.

In our algorithm, the video sequence is categorised into two types: fast encoding frames and threshold updating frames (F_{UPDATE}). The proposed fast CU size decision algorithm is incorporated in the fast encoding frames, and the threshold updating process is implemented in the F_{UPDATE} frames. In each F_{UPDATE} frame, an exhaustive CU size decision is performed to determine the best coding decision. When the encoding of each F_{UPDATE} frame is completed, the R_{CU} values of all encoded LCUs are calculated and the coding depth level of each 4×4 block is stored. The R_{CU} values are subsequently arranged in ascending order $[R_{min}, \dots, R_{max}]$, and R_{med} is the median R_{CU} .

The upper threshold T_U is chosen from the set of $[R_{med}, \dots, R_{max}]$. Each R_{CU} value in the set of $[R_{med}, \dots, R_{max}]$ is considered as a candidate value T_U^c of T_U . The error ratio (R) is used as the criterion by which to select the upper threshold T_U . For each T_U^c , the value of R for the LCUs which have R_{CU} values greater than T_U^c is calculated as:

$$R = E_{cnt} / (E_{cnt} + C_{cnt}) \tag{5}$$

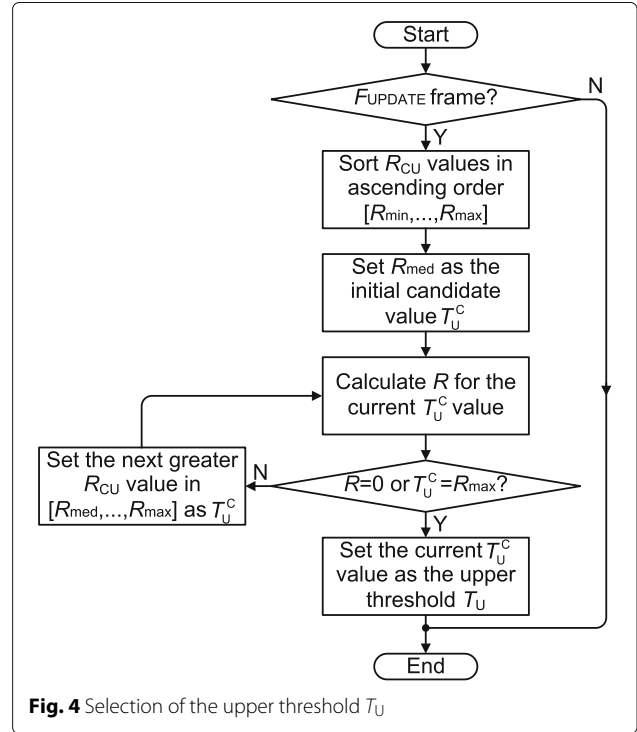
where E_{cnt} denotes the number of 4×4 blocks encoded using coding depth 0 in all the LCUs whose R_{CU} values are greater than T_U^c , and C_{cnt} is the number of 4×4 blocks encoded using a coding depth of 1, 2, or 3. E_{cnt} and C_{cnt} are computed as shown in Eq. (6).

$$E_{cnt} = \begin{cases} E_{cnt} & \text{if } 1 \leq D_i \leq 3 \\ E_{cnt} + 1 & \text{otherwise} \end{cases}, \quad C_{cnt} = \begin{cases} C_{cnt} + 1 & \text{if } 1 \leq D_i \leq 3 \\ C_{cnt} & \text{otherwise} \end{cases} \tag{6}$$

where D_i denotes the coding depth used by the i th 4×4 block. T_U^c traverses from small to large R_{CU} values. This process terminates when R reaches 0, and the current T_U^c value is chosen as the upper threshold T_U . Each of the subsequent fast encoding frames uses this T_U value until it is updated in the next F_{UPDATE} frame. The algorithm for updating T_U is presented in Fig. 4. The lower threshold T_L is determined in a similar way to T_U , but T_L is chosen from $[R_{min}, \dots, R_{med}]$.

4.3 Skipping and early termination of CU sizes

In order to skip unlikely depth levels, the R_{CU} value of each LCU in fast encoding frames is compared with the upper threshold T_U and the lower threshold T_L . If the R_{CU} value is greater than T_U , the current LCU is likely to be located in a region with high spatial detail. Therefore, the range of coding depth levels to be examined is set to [1,3], namely, the depth level 0 is skipped. For the LCUs that have an R_{CU} value less than T_L , a depth range of [0, 2] is



used. Formally, the initial coding depth range DR_0 can be expressed as

$$DR_0 = [d_L, d_U] = \begin{cases} [0, 2] & \text{if } R_{CU} < T_L \\ [1, 3] & \text{if } R_{CU} > T_U \\ [0, 3] & \text{otherwise} \end{cases} \tag{7}$$

where d_U and d_L represent the maximum and minimum depth levels for the current LCU, respectively.

In addition to the AC energy, the spatial correlation is employed to further narrow the range of coding depth levels. As adjacent blocks generally contain similar textures, the best coding depth level of the current LCU is usually strongly correlated to those of the coded neighbouring LCUs. If both the best depth levels of the above and left coded LCUs are not greater than 1, the range of coding depth levels to be examined is reduced to $[d_L, d_U - 1]$. If both the above and left neighbours of the current LCU are coded using depth level 2 or 3, the depth range is narrowed to $[d_L + 1, d_U]$. Otherwise, the maximum and minimum depth levels for the current LCU remain unchanged. Therefore, the depth range DR_1 is given as shown in Eq. (8).

$$DR_1 = [d_L, d_U] = \begin{cases} [d_L, d_U - 1] & \text{if } d_{left} \leq 1, d_{abv} \leq 1 \\ [d_L + 1, d_U] & \text{if } d_{left} > 1, d_{abv} > 1 \\ [d_L, d_U] & \text{otherwise} \end{cases} \tag{8}$$

where d_{abv} and d_{left} denote the best depth levels of the above and left coded LCUs, respectively.

Due to the strong correlation between successive frames, the coding depth range of the current CU can be predicted by using the partitioning information of the co-located LCU in previously coded frames. If the best depth level of the corresponding LCU in the previous frame is greater than $d_L + 1$, the range of coding depth levels to be examined is reduced to $[d_L + 1, d_U]$. If the best depth level of the corresponding LCU is less than $d_U - 1$, the range of coding depth levels to be examined is reduced to $[d_L, d_U - 1]$. Otherwise, the maximum and minimum depth levels for the current LCU remain unchanged. The coding depth range DR_2 is determined as

$$DR_2 = [d_L, d_U] = \begin{cases} [d_L, d_U - 1] & \text{if } d_{prev} < d_U - 1 \\ [d_L + 1, d_U] & \text{if } d_{prev} > d_L + 1 \\ [d_L, d_U] & \text{otherwise} \end{cases} \quad (9)$$

where d_{prev} denotes the depth level of the co-located LCU in the previous frame.

Empirical results from applying our fast CU size decision algorithm are summarised in Fig. 5. The diagram shows the proportion of LCUs for which evaluation on depth level 0 is avoided (SP) and the proportion of LCUs for which early termination is achieved (ETP). The decision accuracy, which indicates the percentage of identical

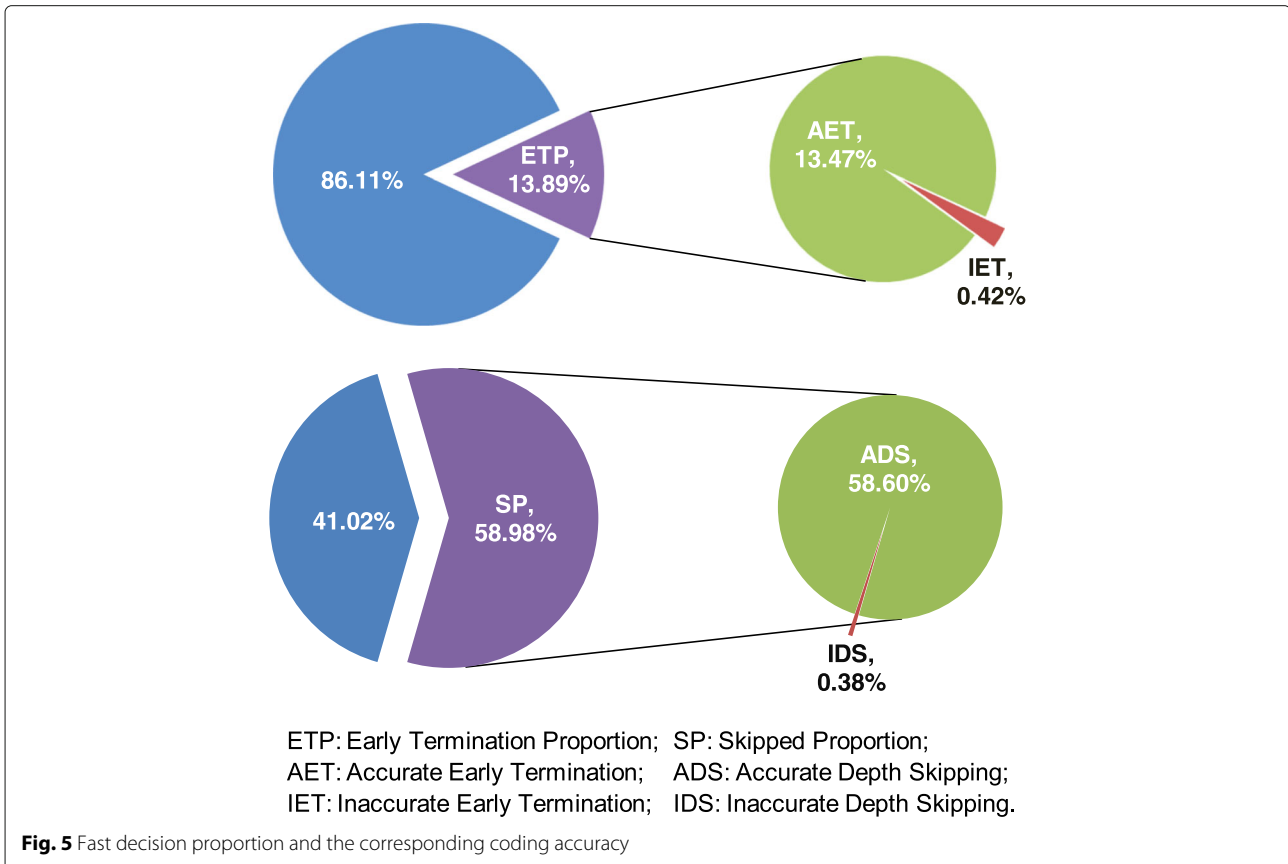
depth decisions generated from a partial evaluation as those generated from an exhaustive evaluation, is also presented in Fig. 5. It can be concluded that the evaluations on depth level 3 can be skipped in advance for a large proportion of LCUs, without affecting the accuracy of the decision-making process. Furthermore, coding depth level 0 can be skipped for the majority of LCUs without causing any noticeable decision-making errors.

5 Fast PU mode decision

HEVC supports up to 35 intra-prediction modes. The increased number of intra-prediction directions improves the encoding accuracy at the expense of a more complicated mode decision process. Consequently, there is a need to improve the intra-prediction mode decision process and thus reduce computational complexity. The method proposed in this section focuses primarily on the coding mode correlation between adjacent quad-tree levels and between temporally neighbouring frames.

5.1 Mode correlation between neighbouring quad-tree levels

In the quad-tree based coding structure, there exists a high-coding mode dependency between the upper and lower coding depth levels. Figure 6 illustrates the possible prediction modes of two neighbouring depth levels, where



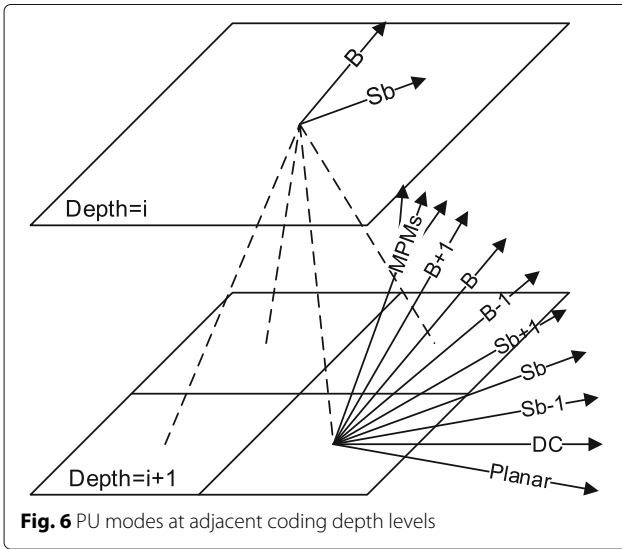


Fig. 6 PU modes at adjacent coding depth levels

B and Sb represent the best and second best coding modes at depth level i . B_n represents the best mode for the co-located PU at level $i+1$. Figure 7 shows the possibility that B_n is selected from various sets of coding modes when B and Sb are both directional modes. The statistical data was obtained from processing five video sequences.

P denotes the possibility that B_n is selected from the set $\{B, Sb, B-1, B+1, Sb-1, Sb+1, \text{planar}, \text{DC}, \text{MPMs}\}$; $P1$ that B_n is from $\{B, B-1, B+1\}$; $P2$ that B_n is from $\{Sb, Sb-1, Sb+1\}$; $P3$ that B_n is a MPM; and $P4$ that B_n is the DC or planar mode. It can be seen from Fig. 7 that there is a high possibility that B_n belongs to the mode set chosen for P . That means if B and Sb are both directional modes, the set of candidate modes for B_n can be reduced to $\{B, Sb, B-1, B+1, Sb-1, Sb+1, \text{planar}, \text{DC}, \text{MPMs}\}$. If the candidate list contains only the most likely modes, the unnecessary computation of unlikely modes can be avoided. It should be noted that any duplicates existing in the overlap of the candidate lists should be removed, thus preventing repeated considerations for any given coding mode.

5.2 Improved candidate mode list

In our algorithm, the least possible prediction modes are not involved in the RMD process. If the current depth level is greater than 0, and both B and Sb of the co-located PU at the previous coding depth level are directional modes, the most likely coding mode for the current PU is predicted by exploiting the mode correlation between neighbouring depth levels in the quad-tree structure. For these PUs, a number of modes are selected from the 35 prediction modes to construct a candidate list CL_1 , as given in Eq. (10); otherwise, the mode information from temporally neighbouring PUs is utilised to construct the candidate list.

$$CL_1 = \begin{cases} \{0, 1, 2, 3, 33, 34\} & \text{if } m_l = 2, m_g = 34 \\ \{0, 1, 2, 3, 4\} & \text{if } m_l = 2, m_g = 3 \\ \{0, 1, 2, 3, 4, 5\} & \text{if } m_l = 2, m_g = 4 \\ \{0, 1, 32, 33, 34\} & \text{if } m_l = 33, m_g = 34 \\ \{0, 1, 31, 32, 33, 34\} & \text{if } m_l = 32, m_g = 34 \\ \left\{ \begin{array}{l} 0, 1, 2, 3, m_g, \\ m_g - 1, m_g + 1 \end{array} \right\} & \text{if } m_l = 2, m_g \in [5, 33] \\ \left\{ \begin{array}{l} 0, 1, m_l - 1, m_l, \\ m_l + 1, 33, 34 \end{array} \right\} & \text{if } m_g = 34, m_l \in [3, 31] \\ CL'_1 & \text{if } B, Sb \in [3, 33] \end{cases} \quad (10)$$

$$CL'_1 = \begin{cases} \left\{ \begin{array}{l} 0, 1, m_l - 1, m_l, \\ m_g, m_g + 1 \end{array} \right\} & \text{if } m_g - m_l = 1 \\ \left\{ \begin{array}{l} 0, 1, m_l - 1, m_l, \\ m_l + 1, m_g, m_g + 1 \end{array} \right\} & \text{if } m_g - m_l = 2 \\ \left\{ \begin{array}{l} 0, 1, m_l - 1, m_l, m_l + 1, \\ m_g - 1, m_g, m_g + 1 \end{array} \right\} & \text{if } m_g - m_l \geq 3 \end{cases} \quad (11)$$

where 0 and 1 represent the planar mode and DC mode, respectively; the others indicate the directional modes; m_l and m_g are defined by Eq. (12).

$$\begin{cases} m_l = B, m_g = Sb & \text{if } B \leq Sb \\ m_l = Sb, m_g = B & \text{otherwise} \end{cases} \quad (12)$$

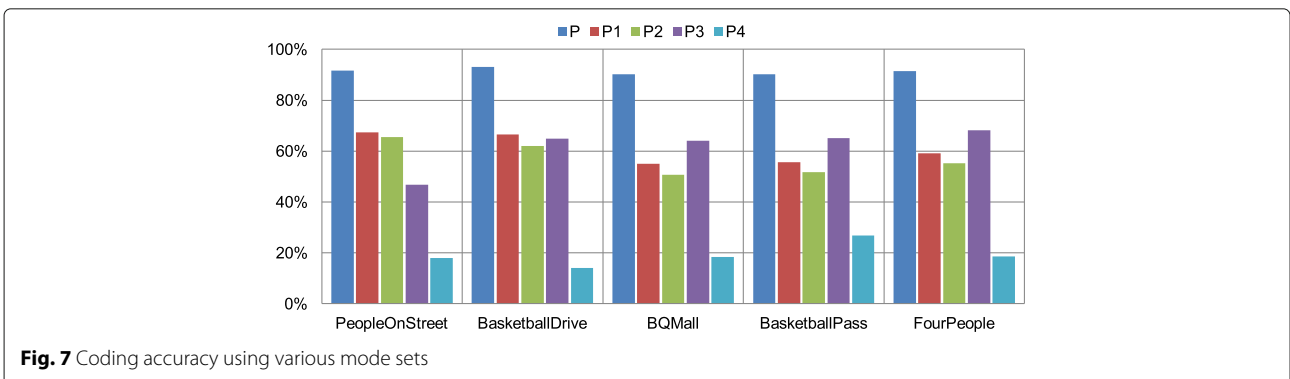


Fig. 7 Coding accuracy using various mode sets

There also exists a strong temporal mode correlation between neighbouring frames. If the co-located PU in the previous frame is coded using a directional mode, the candidate list CL_2 as shown in Eq. (13) is constructed; otherwise, CL_3 is employed, which is given by Eq. (14).

$$CL_2 = \begin{cases} \{0, 1, P-1, P, P+1\} & \text{if } P \in [3, 33] \\ \{0, 1, 2, 3, 4\} & \text{if } P = 2 \\ \{0, 1, 32, 33, 34\} & \text{if } P = 34 \end{cases} \quad (13)$$

$$CL_3 = \begin{cases} \{0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, \} \\ \{20, 22, 24, 26, 28, 30, 32, 34\} \end{cases} \quad (14)$$

where P is the best coding mode of the co-located PU in the previous temporal frame.

The prediction modes that remain in the candidate list are examined in the RMD process. The default HM implementation retains N modes after RMD, whereas only M ($M < N$) modes are retained in our algorithm. The values of N and M are shown in Table 1.

The RMD process is performed only once for the PUs which use CL_1 or CL_2 as the candidate mode list, and subsequently, the M modes with the minimum Hadamard cost ($HCOST_{MODE}$) are selected. For PUs that employ CL_3 as the candidate list, M modes are selected through the first RMD process. The prediction modes next to a directional mode that has been included in the retained M modes are combined with the retained M modes to construct a new candidate mode list CL'_3 . Subsequently, the prediction modes in CL'_3 are evaluated in the second RMD process, and the M modes with the minimum Hadamard cost are finally retained.

The proposed algorithm employs $HCOST_{MODE}$ values to further eliminate the modes in the candidate list. Let $HCOST_{MODE}^{\min}$ denote the minimum value of $HCOST_{MODE}$. The $HCOST_{MODE}$ value of each retained mode is evaluated. If the $HCOST_{MODE}$ value of a candidate mode is greater than $1.5 \times HCOST_{MODE}^{\min}$, this mode is deleted from the candidate list. If there is only one candidate mode left in the candidate list, this mode is determined as the best mode. If there is more than 1 remaining mode in the candidate list, the two modes with the minimum $HCOST_{MODE}$ values are evaluated in the RQT process.

6 Hybrid fast intra-coding algorithm

The two aforementioned algorithms are integrated to form a hybrid fast intra-coding scheme. The overall flowchart of the proposed fast intra-coding algorithm for HEVC is illustrated in Fig. 8. The algorithm can be described as follows:

Table 1 Number of retained modes after RMD for various PU sizes

PU size	4 × 4	8 × 8	16 × 16	32 × 32	64 × 64
N	8	8	3	3	3
M	2	2	2	1	1

1. Check the type of the current frame. If it is a F_{UPDATE} frame, set the coding depth range $[d_L, d_U]$ to $[0, 3]$, and go to step 11. Otherwise, go to step 2;
2. Compare the current LCU's R_{CU} value with T_L . If R_{CU} is less than T_L , set $[d_L, d_U]$ to $[0, 2]$, and go to step 5. Otherwise, go to step 3;
3. Compare the current LCU's R_{CU} value with T_U . If R_{CU} is greater than T_U , set $[d_L, d_U]$ to $[1, 3]$, and go to step 5. Otherwise, go to step 4;
4. Set $[d_L, d_U]$ to $[0, 3]$, and go to step 5;
5. Check the depth levels of the left and above LCUs. If neither of them is greater than 1, set $[d_L, d_U]$ to $[d_L, d_U - 1]$, and go to step 8. Otherwise, go to step 6;
6. Check the depth levels of the left and above LCUs. If they are both greater than 1, set $[d_L, d_U]$ to $[d_L + 1, d_U]$, and go to step 8. Otherwise, go to step 7;
7. Maintain the coding depth range unchanged at $[d_L, d_U]$, and go to step 8;
8. Check the depth level d_{prev} of the co-located LCU in the previous frame. If it is less than $d_L - 1$, set $[d_L, d_U]$ to $[d_L, d_U - 1]$, and go to step 11. Otherwise, go to step 9;
9. Check the depth level d_{prev} of the co-located LCU in the previous frame. If it is greater than $d_U + 1$, set $[d_L, d_U]$ to $[d_L + 1, d_U]$, and go to step 11. Otherwise, go to step 10;
10. Maintain the coding depth range unchanged at $[d_L, d_U]$, and go to step 11;
11. Perform fast intra-mode decision as follows:
 - 11.1 Check the current coding depth, the best mode B and the second best mode Sb of the corresponding PU at the upper level. If the current depth level is not 0, and both B and Sb are directional modes, go to step 11.3. Otherwise, go to step 11.2;
 - 11.2 Check the best coding mode of the co-located PU in the previous frame. If it is a directional mode, go to step 11.4. Otherwise, go to step 11.5;
 - 11.3 Construct CL_1 , and go to step 11.6;
 - 11.4 Construct CL_2 , and go to step 11.6;
 - 11.5 Construct CL_3 , and go to step 11.6;
 - 11.6 Remove the duplicated modes from the candidate list. Perform RMD and retain M prediction modes. For PUs that use CL_3 , go to step 11.7. For PUs that use CL_1 or CL_2 , go to step 11.8;
 - 11.7 Construct CL'_3 , and perform RMD. Retain M candidate modes, and go to step 11.8;
 - 11.8 Combine the retained M prediction modes with MPMs, and evaluate the $HCOST_{MODE}$ values. Remove the modes that have a

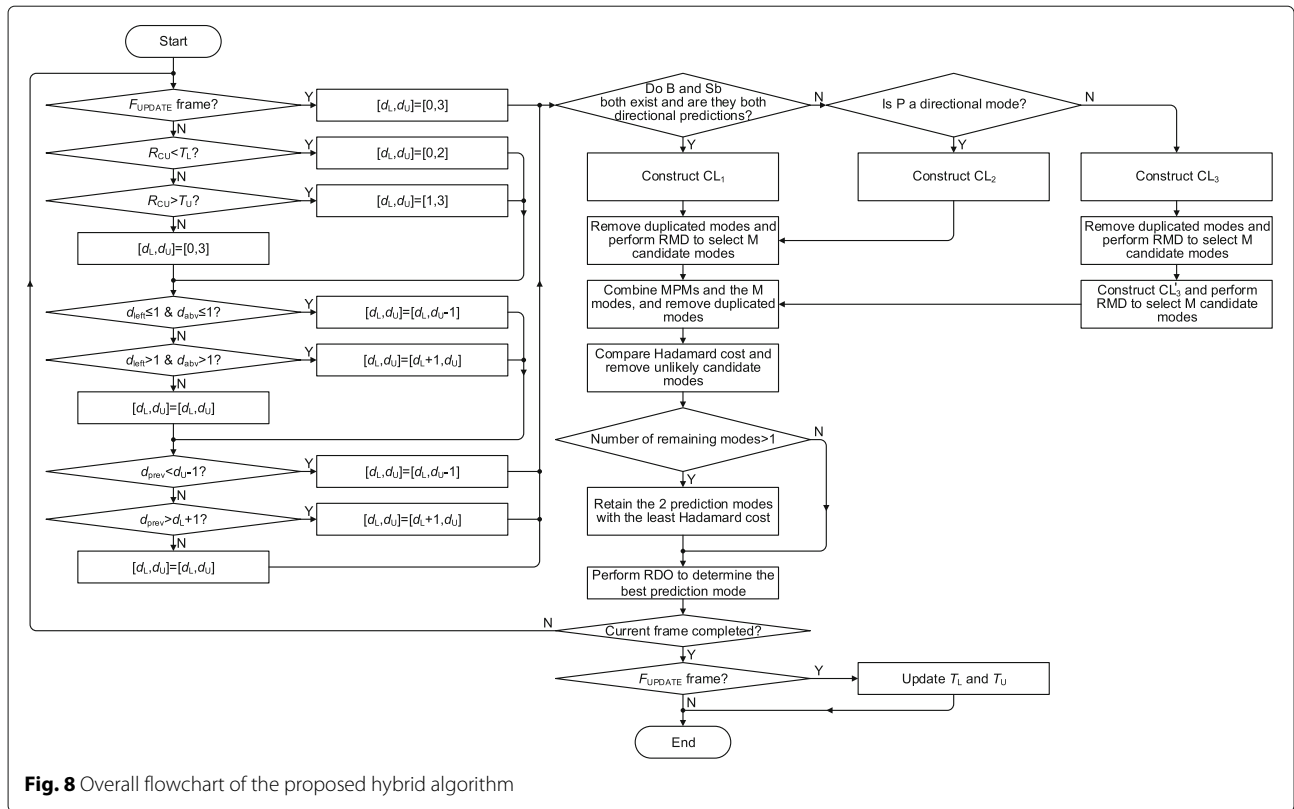


Fig. 8 Overall flowchart of the proposed hybrid algorithm

$HCOST_{MODE}$ value greater than $1.5 \times HCOST_{MODE}^{Min}$, and go to step 11.9;

- 11.9 Check whether the number of remaining modes is greater than 1. If so, go to step 11.10. Otherwise, go to step 11.11;

- 11.10 Retain the 2 prediction modes with the minimum $HCOST_{MODE}$ values, and go to step 11.11;

- 11.11 Perform RQT to select the best coding mode, and go to step 12;

- 12. Check whether the current frame has been completely encoded. If so, go to step 13. Otherwise, go to step 1;
- 13. Check the type of the current frame. If it is a F_{UPDATE} frame, update T_L and T_U . Otherwise, process the next frame.

7 Experimental configuration

To evaluate the performance of the proposed algorithms, all methods were implemented in the HEVC reference software HM 13.0 [28] with the all intra (AI) configuration. Seventeen video sequences [29] with different spatial resolutions (QWVGA, WVGA, 720p, 1080p and WQXGA) were tested, with QP values of 22, 27, 32 and 37. The size of the CTU was fixed to 64×64 , and the

maximum quad-tree depth was set to 4. The simulations were performed on the Microsoft Windows 7 operating system (64 bits) with an Inter Core i5-4590 3.30 GHz CPU and a 4 G RAM. Table 2 presents some typical settings of the encoding parameters, and other parameters used in the simulations are those recommended by the JCT-VC in document JCTVC-K1100 [30].

In order to verify the effectiveness of the algorithms and to perform fair comparison with the standard HM implementation and the fast algorithms recently proposed in [2, 3, 8–10, 14, 16, 20, 21], the performance is measured by the following parameters:

- (1) The Bjøntegaard bit-rate (BDBR, %) and Bjøntegaard PSNR (BDPSNR, dB) as defined in [31];

Table 2 Setting of encoding parameters

Parameter	Value
MaxCUWidth	64
MaxCUHeight	64
MaxPartitionDepth	4
RDOQ	True
AdaptiveQpSelection	False
RecalculateQPAccordingToLambda	False

(2) The encoding time reduction (TR, %), which is calculated as

$$TR = (T_{HM} - T_P) / T_{HM} \times 100\%, \tag{15}$$

where T_{HM} is the encoding time of the HM implementation and T_P is that of the proposed algorithm.

8 Results and discussion

Tables 3 and 4 show the RD performance and encoding time reduction of the proposed fast CU size decision algorithm and the fast coding mode decision algorithm, respectively. Performance comparisons between the proposed hybrid fast intra-coding algorithm and three state-of-the-art methods are given in Table 5. The simulation results for each sequence shown in Tables 3, 4 and 5 are the average values for the given QP factors. The fast algorithms in [2, 3, 8–10, 14, 16, 20, 21] were developed based on different versions of HM. However, the intra-coding performance of these HM versions are very similar, as HEVC intra-coding changed very little between different versions [2, 21]. Therefore, it is reasonable to suppose that it is fair to compare the methods under these different HM versions.

For the fast CU size decision algorithm, it can be seen from Table 3 that the encoding time is significantly

reduced, while incurring a negligible degradation in PSNR and an insignificant increment in bit rate. The simulation results show that our proposed algorithm achieves an average time saving of 29.61% relative to the original HM encoder, with a maximum of 56.77% in ‘Kimono1’ (1920 × 1080 pixels) and a minimum of 18.30% in ‘PartyScene’ (832 × 480 pixels). There is a negligible loss in coding efficiency, with 0–0.04 dB drop in PSNR or 0.02–0.88% increase in bit rate. In addition to the HM implementation, we also compared the proposed algorithm with the state-of-the-art fast CU depth decision scheme for HEVC [2, 3, 8]. A performance comparison with the algorithms proposed by Min, Shen and Huang is also presented in Table 3. Compared with Min’s method, although the proposed algorithm reduced encoding time less, better rate distortion performance was obtained. For the high-resolution sequence ‘Kimono1’ comprising high spatial detail, Min’s algorithm incurred an average increase in bit rate of 3.64%, which is significant. Therefore, the improved time reduction of Min’s algorithm was achieved at the expense of lower coding efficiency. However, our algorithm produces only a negligible increment in bit rate. Specifically, the increase in bit-rate was only 0.88% in the worst case. The degradation in coding efficiency can be regarded as negligible. Compared with Shen’s and Huang’s methods, similar comparative results can be observed. It

Table 3 Comparison of the proposed fast CU size decision algorithm against the existing algorithms and HM 13.0

Class	Resolution	Sequence	Min [2]		Shen [3]		Huang [8]		Proposed	
			BDBR	TR	BDBR	TR	BDBR	TR	BDBR	TR
A	2560 × 1600	Traffic	0.80	57.4	0.80	23.0	0.70	43.2	0.29	27.2
		PeopleOnStreet	0.50	55.9	0.60	30.0	0.70	37.5	0.35	22.9
		BasketballDrive	0.70	52.5	2.80	22.0	0.60	46.8	0.88	38.5
B	1920 × 1080	BQTerrace	0.40	45.3	0.60	17.0	0.50	42.3	0.24	28.0
		Cactus	0.60	47.7	0.60	19.0	0.80	42.9	0.29	27.7
		Kimono1	3.60	57.4	0.50	21.0	1.20	50.9	0.40	56.8
		ParkScene	0.60	48.0	0.70	26.0	0.80	43.8	0.42	28.8
C	832 × 480	BasketballDrill	1.00	50.7	0.60	17.0	0.60	36.1	0.34	22.7
		BQMall	0.60	51.9	0.40	22.0	0.80	33.9	0.26	24.2
		PartyScene	0.00	40.2	0.00	19.0	0.40	31.3	0.02	18.3
		RaceHorses	0.20	48.9	0.30	21.0	0.50	38.9	0.56	25.6
		BasketballPass	0.60	50.5	0.40	21.0	0.20	33.3	0.27	28.0
D	416 × 240	BlowingBubbles	0.20	42.2	0.00	24.0	0.10	24.9	0.05	26.3
		BQSquare	0.00	42.8	0.10	22.0	0.00	30.1	0.03	28.7
		RaceHorses	0.50	41.1	0.10	25.0	0.20	26.8	0.20	28.1
E	1280 × 720	FourPeople	0.70	58.5	0.50	23.0	0.80	38.7	0.24	28.3
		Johnny	1.30	65.1	2.00	26.0	0.80	47.1	0.38	43.3
—	—	Average	0.68	49.4	0.56	22.0	0.56	37.6	0.30	28.8

Table 4 Comparison of the proposed fast coding mode decision algorithm against the existing algorithms and HM 13.0

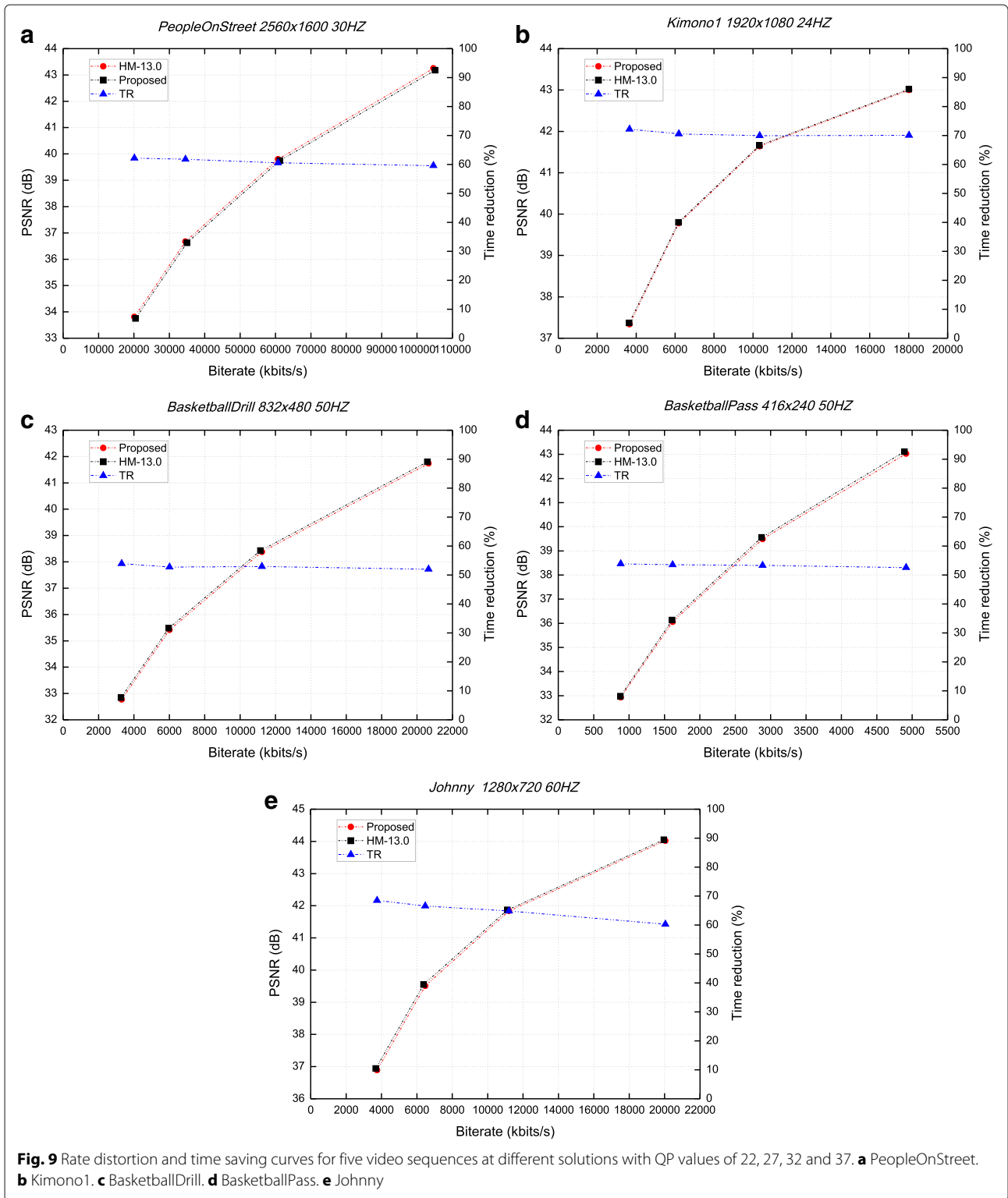
Class	Resolution	Sequence	Jamali [10]		Xiang [14]		Chen [16]		Proposed	
			BDBR	TR	BDBR	TR	BDBR	TR	BDBR	TR
A	2560 × 1600	Traffic	0.95	35.4	0.33	25.0	0.88	24.0	1.39	36.0
		PeopleOnStreet	1.00	34.1	0.45	26.0	1.07	24.0	1.85	50.6
		BasketballDrive	2.17	38.4	0.41	25.0	1.73	23.7	1.15	39.3
B	1920 × 1080	BQTerrace	0.79	35.2	0.52	28.0	0.98	22.1	1.20	42.3
		Kimono1	0.79	39.2	0.10	26.0	0.63	24.5	0.45	35.8
		Cactus	1.34	36.1	0.41	23.0	1.37	23.7	1.42	37.7
		ParkScene	0.87	37.5	0.37	24.0	0.41	19.8	0.79	38.9
C	832 × 480	BasketballDrill	0.80	33.1	0.54	25.0	1.53	31.3	1.79	39.0
		BQMall	1.15	34.3	0.70	25.0	1.30	24.6	1.73	38.8
		PartyScene	1.18	32.8	0.81	26.0	1.26	23.0	1.90	37.3
		RaceHorses	0.72	34.7	0.39	22.0	0.86	30.8	1.06	36.4
D	416 × 240	BasketballPass	1.45	36.0	0.69	28.0	-	-	1.65	36.7
		BlowingBubbles	1.01	33.7	0.82	24.0	1.23	22.6	1.49	36.9
		BQSquare	1.38	32.7	0.86	24.0	-	-	2.45	37.4
E	1280 × 720	RaceHorses	0.99	34.1	0.69	25.0	1.11	22.2	1.64	39.6
		FourPeople	-	-	-	-	-	-	1.85	39.8
		Johnny	-	-	-	-	1.83	33.7	1.87	40.2
—	—	Average	1.11	35.2	0.54	25.1	1.16	25.0	1.51	39.0

Table 5 Comparison of the proposed hybrid algorithm against the existing algorithms and HM 13.0

Class	Resolution	Sequence	Gao [9]		Shang [20]		Zhang [21]		Proposed	
			BDBR	TR	BDBR	TR	BDBR	TR	BDBR	TR
A	2560 × 1600	Traffic	0.90	27.3	0.37	38.6	0.78	55.4	1.47	54.4
		PeopleOnStreet	0.90	24.7	0.41	37.1	0.92	53.7	1.99	61.0
		BasketballDrive	-	-	-	-	0.77	65.1	1.87	60.1
		BQTerrace	-	-	0.69	40.4	0.61	53.4	1.24	57.4
B	1920 × 1080	Kimono1	1.30	24.1	0.28	43.1	0.66	65.8	0.83	70.7
		Cactus	1.00	24.2	0.71	38.2	0.72	54.0	1.52	54.7
		ParkScene	0.70	26.1	-	-	0.70	54.1	1.09	56.3
		BasketballDrill	1.20	28.6	0.77	36.9	0.84	45.0	1.89	52.9
C	832 × 480	BQMall	0.90	34.2	0.91	37.6	0.59	50.8	1.73	52.2
		PartyScene	0.70	27.3	-	-	0.63	39.4	1.64	48.7
		RaceHorses	-	-	-	-	0.68	51.7	1.46	52.0
		BasketballPass	1.20	32.5	0.84	35.2	0.77	50.7	1.62	53.3
D	416 × 240	BlowingBubbles	-	-	0.83	35.8	0.45	40.6	1.23	46.4
		BQSquare	1.10	26.7	-	-	0.74	44.0	2.18	48.6
		RaceHorses	1.00	29.3	-	-	0.57	44.3	1.58	49.5
E	1280 × 720	FourPeople	1.10	22.3	0.71	38.3	0.96	60.3	1.85	55.5
		Johnny	1.30	29.7	0.55	39.7	1.18	67.2	2.06	65.1
—	—	Average	1.02	27.5	0.64	38.3	0.74	52.7	1.60	55.2

can be concluded that the proposed fast CU size decision algorithm reduced encoding time less, but higher coding efficiency was maintained. Therefore, the selection of fast CU size decision algorithm depends on the specific application scenarios and the user's requirements.

For the fast coding mode decision algorithm, Table 4 shows that the average time reduction is 38.98%, while BDBR increases by 1.51% and BDPSNR decreases by 0.09 dB. As shown in Table 4, our algorithm consistently results in a significant reduction in computational



complexity, while keeping nearly the same RD performance as that of the HM encoder. The experimental results indicate that the algorithm performs well for all types of video sequences. This result verifies that there exists strong mode correlation between coding depth levels and between neighbouring frames. Once the correlation is fully exploited, the unlikely coding modes can be skipped. The comparative results of the proposed algorithm and the fast mode decision algorithm proposed by Jamili [10], Xiang [14] and Chen [16] are also included in Table 4. It can be seen that our algorithm saves more than 3, 14 and 14% in average encoding time compared to Jamili's, Xiang's and Chen's methods, while achieving very similar coding efficiency. Our proposed algorithm reduces more encoding time than the existing methods for all types of video sequence. Therefore, the proposed fast coding mode decision algorithm provides the best performance in terms of encoding time reduction.

Table 5 provides the simulation results of the proposed hybrid algorithm when compared with the original HM implementation and three state-of-the-art fast HEVC intra-coding algorithms, namely, those proposed by Gao [9], Shang [20] and Zhang [21]. It can be seen that the time reduction achieved by our algorithm is 55.24% on average. The maximum time reduction is 70.68% for the sequence of 'Kimono1', which contains significant detail. This is because the evaluation of unlikely depth levels is effectively avoided. On the other hand, the encoding time reduction is achieved by no more than a 2.18% increase in bit rate. Compared with Gao, Shang and Zhang's methods, the proposed algorithm saves additional time of 28, 17 and 3% with similar rate distortion performance. It can be observed from Table 5 that, although the coding efficiency of Zhang's method is slightly higher than our algorithm, its time reduction is less. Therefore, the comparative results demonstrate that our hybrid method outperforms the algorithms proposed by Gao and Shang and achieves a comparable performance to Zhang's method. Specifically, when encoding low (416×240 pixels) and medium (832×480 pixels) resolution video sequences, our algorithm consistently outperforms Zhang's method in terms of encoding time reduction. When encoding high definition (HD, 1280×720 pixels), full HD (FHD, 1920×1080 pixels) and 2K (2560×1600 pixels) video sequences, the relative coding performance is affected by the content of the video test sequences. For video sequences that contain rich detail, such as 'PeopleOnStreet' and 'Kimono1', our algorithm shows better performance than Zhang's. Therefore, the proposed algorithm provides a better choice for high resolution sequences containing complex detail.

Figure 9 shows the RD curves for five video sequences with different spatial resolutions under a variety of QP factors. The corresponding time saving curves are also

presented on the same diagrams. It can be seen that the RD curves of the proposed hybrid algorithm almost overlap with those of the HM encoder, which means the RD performance of the proposed algorithm is very similar to that of the HM benchmark. Therefore, there is nearly no quality loss under a wide range of bit rates. The time saving curves show that the hybrid algorithm consistently achieves a time reduction of more than 46% for different video sequences. In summary, our proposed algorithm significantly reduces the encoding time with negligible effect on bit rate and picture quality.

9 Conclusions

This paper focused on developing a computational complexity reduction scheme for HEVC intra-coding. We combined a fast CU size decision algorithm and a fast intra-prediction mode decision algorithm to form a hybrid scheme. The proposed fast CU size decision algorithm employed the homogeneity of video content, the partitioning information of spatially adjacent coded PUs and temporally neighbouring frames to gradually narrow the depth range. In the fast mode decision algorithm, the coding mode correlation between coding depth levels was utilised to eliminate candidate modes in the candidate list, and the number of candidate modes was further reduced by evaluating the Hadamard cost of the remaining modes. The performance was compared with the original HEVC encoder as well as previously proposed complexity reduction schemes. Simulation results showed that the proposed algorithm outperformed existing schemes, achieving up to 70.68% reduction in coding time, while barely degrading the picture quality and bit rate.

Abbreviations

AC: Alternating current; AI: All intra; BDBR: Bjøntegaard bitrate; BDPSNR: Bjøntegaard PSNR; CTU: Coding tree unit; CU: Coding unit; FHD: Full HD; HD: High definition; HEVC: High efficiency video coding; IBC: Intra block copy; JCT-VC: Joint collaborative team on video coding; LCU: Largest CU; MB: Macroblock; MPM: Most probable mode; PU: Prediction unit; RD: Rate distortion; RDO: Rate distortion optimisation; RMD: Rough mode decision; RQT: Residual-quad-tree; SCU: Smallest CU; SVM: Support vector machine; TU: Transform unit; TR: Time reduction

Acknowledgements

The authors would like to thank Professor Graham R. Martin in the Department of Computer Science at the University of Warwick for his very helpful suggestions. The authors also would like to thank the editor and anonymous reviewers for their valuable comments.

Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under project no. 61401123, the Fundamental Research Funds for the Central Universities under grant no. HIT.NSRIF.201617 and the Harbin Science and Technology Bureau under project no. 2014RFQXJ166. The above funding bodies provided financial support only.

Availability of data and materials

The dataset supporting the conclusions of this article is available in the [H.265/HEVC reference software] repository [https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/].

Authors' contributions

XL designed the algorithms and drafted the manuscript. CY performed the experiments and analysed the data. XJ reviewed and revised the whole paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

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

Author details

¹School of Electronics and Information Engineering, Harbin Institute of Technology, No. 92 West Dazhi Street, Nangang District, Harbin, China, 150001. ²School of Computer and Information Engineering, Harbin University of Commerce, No.1 Xuehai Street, Songbei District, Harbin, China, 150028.

Received: 13 January 2018 Accepted: 30 May 2018

Published online: 18 June 2018

References

- G Sullivan, J Ohm, W Han, T Wiegand, Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
- B Min, R Cheung, A fast CU size decision algorithm for the HEVC intra encoder. *IEEE Trans. Circuits Syst. Video Technol.* **25**(5), 892–896 (2015)
- L Shen, Z Zhang, Z Liu, Effective CU size decision for HEVC intracoding. *IEEE Trans. Image Process.* **23**(10), 4232–4241 (2014)
- H Zhang, Q Zhou, N Shi, F Yang, X Feng, Z Ma, in *Proc. IEEE Int Conf Acoust Speech Signal Proc. Fast intra mode decision and block matching for HEVC screen content compression* (IEEE, Shanghai, China, 2016), pp. 1377–1381
- L Shen, Z Liu, X Zhang, W Zhao, Z Zhang, An effective CU size decision method for HEVC encoders. *IEEE Trans. Multimedia.* **15**(2), 465–470 (2013)
- S Cho, M Kim, Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. *IEEE Trans. Circuits Syst. Video Technol.* **23**(9), 1555–1564 (2013)
- F Mu, L Song, X Yang, Z Luo, in *Proceedings of IEEE International Conference on Multimedia and Expo Workshops. Fast coding unit depth decision for HEVC* (IEEE, Chengdu, China, 2014), pp. 1–6
- X Huang, H Jia, K Wei, J Liu, C Zhu, Z Lv, D Xie, in *Proceedings of IEEE International Conference on Visual Communications and Image Processing. Fast algorithm of coding unit depth decision for HEVC intra coding* (IEEE, Valletta, Malta, 2014), pp. 458–461
- L Gao, S Dong, W Wang, R Wang, in *Proceedings of IEEE International Symposium on Circuits and Systems. Fast intra mode decision algorithm based on refinement in HEVC* (IEEE, Lisbon, Portugal, 2015), pp. 517–520
- M Jamali, S Coulombe, F Caron, in *Proceedings of Data Compression Conference. Fast HEVC intra mode decision based on edge detection and SATD costs classification* (IEEE, Snowbird, UT, USA, 2015), pp. 43–52
- T Silva, L Agostini, L Cruz, in *Proceedings of European Signal Processing Conference. Fast HEVC intra prediction mode decision based on edge direction information* (IEEE, Bucharest, Romania, 2012), pp. 1214–1218
- V Sanchez, in *Proceedings of IEEE Global Conference on Signal and Information Processing. Fast intra-prediction for lossless coding of screen content in HEVC* (IEEE, Orlando, FL, USA, 2015), pp. 1367–1371
- X Lu, G Martin, Fast mode decision algorithm for the H.264/AVC scalable video coding extension. *IEEE Trans. Circuits Syst. Video Technol.* **23**(5), 846–855 (2013)
- W Xiang, C Cai, Z Wang, H Zeng, J Chen, in *Proceedings of Tenth International Conference on Signal-Image Technology and Internet-based Systems. Fast intra mode decision for HEVC* (IEEE, Marrakech, Morocco, 2014), pp. 283–288
- S Na, W Lee, K Yoo, in *Proceedings of IEEE International Conference on Consumer Electronics. Edge-based fast mode decision algorithm for intra prediction in HEVC* (IEEE, Las Vegas, NV, USA, 2014), pp. 11–14
- G Chen, Z Liu, T Ikenaga, D Wang, in *Proceedings of IEEE International Symposium on Circuits and Systems. Fast HEVC intra mode decision using matching edge detector and kernel density estimation alike histogram generation* (IEEE, Beijing, China, 2013), pp. 53–56
- S Lim, H Kim, Y Choi, S Yu, Fast intra-mode decision method based on DCT coefficients for H.264/AVC. *Signal Image and Video Process.* **9**(2), 481–489 (2015)
- H Zhang, Z Ma, Fast intra mode decision for high efficiency video coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.* **24**(4), 660–668 (2014)
- N Hu, E Yang, Fast mode selection for HEVC intra-frame coding with entropy coding refinement based on a transparent composite model. *IEEE Trans. Circuits Syst. Video Technol.* **25**(9), 1521–1532 (2015)
- X Shang, G Wang, T Fan, Y Li, in *Proceedings of IEEE International Conference on Image Processing. Fast CU size decision and PU mode decision algorithm in HEVC intra coding* (IEEE, Quebec City, QC, Canada, 2015), pp. 207–213
- T Zhang, M Sun, D Zhao, W Gao, Fast intra-mode and CU size decision for HEVC. *IEEE Trans Circuits Syst. Video Technol.* **27**(8), 1714–1726 (2017)
- L Zhao, X Fan, S Ma, D Zhao, Fast intra-encoding algorithm for high efficiency video coding. *Signal Process. Image Commun.* **29**(9), 935–944 (2014)
- J Lei, D Li, Z Pan, Z Sun, S Kwong, C Hou, Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding. *IEEE Trans. Broadcasting.* **63**(1), 48–58 (2017)
- L Wang, W Siu, Novel adaptive algorithm for intra prediction with compromised modes skipping and signaling processes in HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **23**(10), 1686–1694 (2013)
- 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)
- D Zhang, Y Chen, E Izquierdo, Fast intra mode decision for HEVC based on texture characteristic from RMD and MPM. *Proceedings of IEEE International Conference on Visual Communications and Image Processing*, 510–513 (2014)
- F Bossen, B Bross, K Suhring, D Flynn, HEVC complexity and implementation analysis. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1685–1696 (2012)
- H.265/HEVC reference software (HM 13.0) and manual. https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware. Accessed 03 Jan 2018
- H.265/HEVC standard test video sequences. <ftp://ftp.tnt.uni-hannover.de/testsequences>. Accessed 03 Jan 2018
- F Bossen, *Common test conditions and software reference configurations. JCT-VC, 11th Meeting.* (JCT-VC, Shanghai, China, 2012). JCTVC-K1100
- G Bjontegaard, *Calculation of average PSNR differences between RD-curves. VCEG, 13th Meeting.* (ITU-T VCEG, Austin, TX, USA, 2001). VCEG-M33

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com