J. Vis. Commun. Image R. 43 (2017) 77–88

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Rough mode cost–based fast intra coding for high-efficiency video coding ☆

Zong-Yi Chen, Pao-Chi Chang *

National Central University, Department of Communication Engineering, No. 300, Jhongda Rd., Taoyuan City 32001, Taiwan

ARTICLE INFO

Article history:
Received 24 August 2016
Revised 8 November 2016
Accepted 17 December 2016
Available online 24 December 2016

Keywords:
High-efficiency video coding (HEVC)
Fast intra coding
Coding unit (CU)
Intra mode decision
Rough mode decision
Transform unit (TU)

ABSTRACT

The quadtree-based coding unit (CU) and transform unit (TU) structure, as well as various prediction units (PUs) of HEVC, considerably increase encoding complexity in intra coding and inter coding. This paper proposes a rough mode cost (RMC)-based algorithm for accelerating CU/TU depth decisions and PU mode decisions in HEVC intra coding. For CU depth decisions, RMC values are used for the fast determination of CU partition. In the case of PU mode decisions, modes with higher RMCs are removed from the candidate list to reduce the number of test modes. For TU depth decisions, the TU partition of the mode with the least RMC is used to determine the TU partitions of remaining modes. The proposed TU partitioning method demonstrates superior performance to the default method in reference software. The proposed algorithm can reduce encoding time by approximately 51% on average, with a 0.69% increase in the Bjøntegaard-Delta (BD) rate.

1. Introduction

With the increasing demand for high-quality and high-resolution video content, in 2010 the Joint Collaborative Team on Video Coding (JCT-VC) started work on establishing a new video coding standard called high-efficiency video coding (HEVC) [1]. HEVC was aimed at halving the bit rate of H.264/AVC [2] while maintaining the same subjective video quality; it was officially finalized as an international standard in 2013. The objective of video coding is to compress the source data into a lower bitrate without excessively sacrificing video quality. Thus, recent video coding technologies are often accompanied by complicated prediction processes, leading to considerably increased encoding complexity.

The HEVC encoder consists of a coding unit (CU), prediction unit (PU), and transform unit (TU). HEVC adopts a quadtree-based CU structure to provide high flexibility for encoding an area with smooth or complex content. The CU is the basic encoding component; it includes PUs and TUs and is similar to the macroblocks (MBs) used in H.264/AVC. The CU size can vary from 64×64 to 8×8 pixels, corresponding to four depths from 0 to 3. In general, a large CU is favored for reducing side information in homogeneous regions. By contrast, a small CU is preferred for preserving texture details in nonhomogeneous regions. Each CU contains one to four PUs, and the PU size is limited to that of the CU. The PU is the basic prediction component, and is used when signaling all information related to prediction (e.g., motion vectors and prediction modes). The TU executes transform, quantization, and entropy coding processes. In contrast to previous video coding standards that have involved the use of a fixed-size transform, the TU of HEVC adopts a residual quadtree (RQT), which features nested quadtree-based transform coding. The quadtree structure uses variable transform block sizes ranging from 32×32 to 4×4, with a maximum depth of three, to adapt to the various characteristics of prediction blocks. Table 1 presents the relationship between CU depth and TU size.

In HEVC intra coding, five PU sizes are available: 64×64, 32×32, 16×16, 8×8, and 4×4. For each PU size, the HEVC encoder supports up to 35 prediction modes: DC, planar, and 33 angular modes [3]. To accelerate the selection of an intra mode, a method called rough mode decision (RMD) [4] is adopted in reference software. First, the rough mode cost (RMC), J_{RMD}, is calculated for all intra modes. The RMC is a simplified measure of the rate-distortion (RD) cost, and it is defined as in Eq. (1):

$$J_{RMD} = SATD + \lambda_{\text{pred}} R_{\text{pred}} \tag{1}$$

where $SATD$ is the sum of the absolute Hadamard-transformed coefficients of the differences between the original and predicted signals, λ_{pred} is a Lagrange multiplier, and R_{pred} is the number of bits required for encoding the intra mode bit (without residuals). The

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail address: pcchang@ce.ncu.edu.tw (P.-C. Chang).

http://dx.doi.org/10.1016/j.jvcir.2016.12.007
1047-3203/

Table 1
Relationship between CU depth and TU size.

CU depth/size	TU size		
	Depth 0	Depth 1	Depth 2
0/64 × 64		32 × 32	16 × 16
1/32 × 32	32 × 32	16 × 16	8 × 8
2/16 × 16	16 × 16	8 × 8	4 × 4
3/8 × 8	8 × 8	4 × 4	

first N small RMC modes are selected as candidates for full rate-distortion optimization (RDO), as given by Eq. (2):

$$J_{FRD} = SSE + \lambda_{\text{mode}} R_{\text{mode}} \tag{2}$$

where SSE is the sum of the squared error between the original and reconstructed signals, λ_{mode} is a Lagrange multiplier and $\lambda_{\text{pred}} = \sqrt{\lambda_{\text{mode}}}$, and R_{mode} is the total number of bits required to encode the intra mode. The value of N is 3 for 64 × 64, 32 × 32, and 16 × 16 PUs, and it is 8 for 8 × 8 and 4 × 4 PUs. Subsequently, the maximum number of the three most-probable modes (MPMs) from the top and the left blocks is included in the candidate list for full RDO if the MPMs do not belong to the N RMD modes. At the full RDO stage, a fast intra RQT method is applied in the default scheme (HHI_RQT_INTRA_SPEEDUP = 1) [5]. Instead of testing all TU depths, only the maximum allowed TU size is applied to each intra mode in the candidate list. Once determined, the optimal mode is tested with full TU depths to determine the TU partition. The intra coding procedure is summarized in Fig. 1.

HEVC encoders execute RDO to calculate the RD costs of all CU sizes to determine the most favorable CU partition, and all possible PU modes/TU sizes are tested at each CU depth. Compared with previous standards, the number of possible testing candidates increases substantially in HEVC, and choosing the optimal combination by RDO is computationally intensive. However, for real-time video encoding systems or power-constrained mobile devices, complexity reduction is required. Therefore, increasingly fast algorithms have been proposed for reducing the encoding complexity of HEVC.

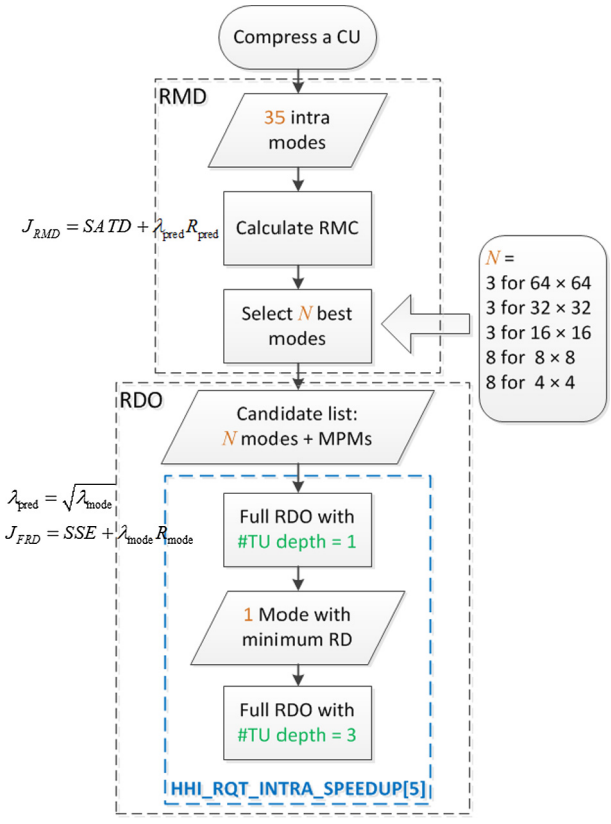


Fig. 1. Flowchart of intra coding in HEVC.

This paper concentrates on accelerating intra coding in HEVC. A fast RMC-based algorithm covering the CU/TU depth decision and PU intra mode decision is proposed. The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the proposed algorithm in detail. The experimental results are described in Section 4, and our concluding remarks are presented in Section 5.

2. Literature review

Many fast algorithms have been proposed for HEVC intra coding. In this paper, we classify such algorithms into three types: CU depth/PU size decisions, PU intra mode decisions, and TU depth decisions.

2.1. CU depth/PU size decisions

Encoding complexity can be lowered considerably by eliminating candidate CU depths (or PU sizes). Most fast CU decision algorithms achieve the early termination (ET) and early splitting (ES) of CUs. ET involves stopping the current CU from splitting into four sub-CUs and testing only the current CU in the RDO process. ES entails splitting the current CU into four sub-CUs directly and skipping the RD cost calculation of the current CU.

Numerous fast CU splitting termination algorithms have been developed by analyzing the content complexity of the current CU. A block with complex texture is often encoded with a small CU. The gradient of a CU has been used to measure block complexity [6,7]. In addition, the mean absolute deviation (MAD) of a CU has been employed as an efficient measure of texture homogeneity [6,8]. Shen et al. calculated the MAD in horizontal and vertical directions to enhance block homogeneity inspection [8]. The pixel variance of the current CU was calculated for comparison with a threshold, and the CU with a small variance was stopped at the current depth [9]. If the variance was greater than the threshold, ES was performed. Zhang et al. classified the complexity of CUs into three categories, namely compound, homogeneous, and undetermined, according to the weighted variance of the scaled SATDs of four sub-CUs, and they performed ES or ET depending on the category [10]. A previous study divided the largest CU (LCU) into 4 × 4 blocks [11]. Starting from the 4 × 4 blocks, if the variance of four subblocks was less than a threshold, these subblocks were merged into a larger block. This merging procedure was repeated and a map of PU locations and sizes was generated. Recursive PU partitioning was avoided, thus reducing computational complexity. Similar to [11], Shi et al. calculated the edge information for block merging to determine a reasonable PU size [12]. Global and local edge complexities in four directions were proposed and used for determining CU partitions [13]. A texture analysis method based on the variation of a pixel with respect to its local neighborhood was proposed for intra CU size selection [14]. Some studies have adopted special feature descriptors in image processing for CU size selection [15,16]. One study selected the local binary pattern operator as the texture descriptor to determine whether a CU should be split [15]. Geuder et al. applied a modified version of the binary robust independent elementary features descriptor to determine CU splitting [16].

On the basis of their strong correlations in the spatial domain, neighboring CU depths have been used to predict the possible range of CU depth for encoding [7,8,17]. Studies have proposed a

CU splitting and pruning method based on the depth information of neighboring CUs [18,19].

In addition to CU texture and neighboring depth information, RD costs have been employed frequently to develop sophisticated and fast algorithms. A fast CU decision method was proposed by following a Bayes' decision rule based on full and low-complexity RD costs [20]. The relationships between the RMC of the current PU and neighboring PUs were studied and used for ES and ET [21]. Moreover, a statistical model-based CU ET approach contingent on RD cost distribution, video content, and quantization parameters (QPs) was presented [22]. Tseng and Lai proposed a fast CU depth decision algorithm based on RD cost and pixel deviation in LCUs [23]. Another study also proposed an ET algorithm based on the RD cost estimated using the aggregated RD costs of sub-CUs [24].

2.2. PU intra mode decisions

For intra mode decisions, the most common method of accelerating the encoder entails reducing the number of modes for RMD and the number of full RDO candidates.

Shang et al. exploited the correlation between the first three RMD modes and higher layer modes to reduce the number of prediction modes [19]. Another study used the intra modes of neighboring PUs to reduce the number of RMD modes [23]. The number of N modes for full RDO was reduced from 3, 3, 3, 8, and 8 to 1, 2, 2, 3, and 3 for PU sizes of 64×64, 32×32, 16×16, 8×8, and 4×4, respectively. The modes that were rarely used in the parent or spatially nearby CUs were skipped in the current CU [17]. Palomino et al. employed a bottom-up encoding order for the CU and used the intra modes of low-level PUs as a reference for selecting the intra modes of the current PU [25].

To avoid the need to directly search 35 intra modes in order, researchers have developed algorithms that test the intra modes in multiple steps, similar to a step search in motion estimation. A previous study presented a mode decision algorithm, called hierarchical mode decision (HMD) [26], that first tests intra modes with frequent directions and then selects the following modes for testing on the basis of previous results. Zhang and Ma proposed a progressive rough mode search (pRMS) that applies a group search method to reduce the number of modes for RMD [24]. Gao et al. defined an optimal adjacent mode (OAM) list consisting of prediction directions similar to those of H.264/AVC to reduce the number of candidates for full RDO [27]. The MPM algorithm was also improved to fully use the spatial correlation between neighboring blocks.

For areas with rare texture information, fewer prediction modes are sufficient. The number of intra prediction modes for RMD was reduced on the basis of the gradient variance of the PU, and a modified RMC calculation approach was proposed for improving the mode selection for full RDO [28]. Tariq et al. used a quadratic model to replace the RMD process by predicting the RD costs of each intra mode according to the corresponding sum of absolute differences [29]. On the basis of a statistical analysis, some modes were excluded from full RDO according to RMC values [30].

2.3. TU depth decisions

Few algorithms are specifically designed for intra TU depth decisions; most existing fast TU depth decision algorithms are applicable to both intra and inter coding. Because a fast intra RQT method is built into reference software, the effectiveness of the existing TU depth decision algorithms is mostly limited.

Quantized transform coefficients are useful for fast TU partition. The number of nonzero coefficients of the root TU was used to terminate subtree RQT processes in HEVC [31]. A block with few nonzero coefficients is highly close to a zero block, and the RD cost of this block is also close to the optimal RD cost. A zero block means that all coefficients in a prediction block are zeroes after quantization. If the number of nonzero coefficients is lower than three, the TU stops splitting [31]. A previous study proposed an early TU splitting termination scheme based on the quasi-zero-block (QZB) concept [32]. The QZB was determined by the sum of all absolute quantized transform coefficients and the number of nonzero coefficients. When coefficients of a block are concentrated at low frequencies, this type of block is well predicted and no further splitting is required. The position of the last nonzero transform coefficient and the number of zero transform coefficients were used to measure the degree of coefficient concentration and determine whether a TU should be split [33]. Instead of using the transformed coefficients, the zero-block detection method in H.264/AVC was extended to HEVC to ensure that it could adapt to various transform sizes; researchers have predicted that the block would theoretically be a zero block on the basis of residuals in the pixel domain [34,35].

In addition to using coefficients, researchers have applied other useful information to determine the ES and ET of TUs. Teng et al. proposed a fast RQT mode decision algorithm for HEVC by considering RD efficiency [36]. A property called zero-block inheritance was observed in [36]; it indicates that if a TU is a zero block, all of its four sub-TUs are likely zero blocks as well. The maximum search depth of intra TUs was reduced in [37] according to PU size. Similar to the CU partitioning strategy, the depth information of the neighboring TUs was used in [38] to predict the depth range of the current TU.

Table 2
Simulation environment for training.

Reference software	HM 15.0 [40]	
Configuration	Intra_main	
Training sequences	ClassA_Traffic	2560 × 1600 @ 30 fps
	CalssB_ParkScene	1920 × 1080 @ 24 fps
	ClassB_Cactus	1920 × 1080 @ 50 fps
	ClassC_BasketballDrill	832 × 480 @ 50 fps
	ClassC_BQMall	832 × 480 @ 60 fps
	ClassD_BQSquare	416 × 240 @ 60 fps
	ClassD_RaceHorses	416 × 240 @ 30 fps
	ClassE_FourPeople	1280 × 720 @ 60 fps
QP	22, 27, 32, 37	
FramesToBeEncoded	10	
Hardware		
CPU	Intel(R) Core(TM) i7-2600 @ 3.40 GHz	
RAM	4.0 G bytes	

Table 3
Mean of $J_{RMD,\ 0}$ for nonsplit and split CUs.

		Depth 0	Depth 1	Depth 2
QP 27	Nonsplit	32,159	8823	1916
	Split	72,139	16,621	4196
QP 37	Nonsplit	31,692	9298	2310
	Split	73,929	18,070	5305

3. Proposed RMC-based fast intra coding

Although numerous fast intra coding algorithms have been proposed, no study has proposed an integrated algorithm for intra CU/PU/TU. In this study, in addition to the characteristics of images (i.e. complexity and gradient), RMC is used as the core feature throughout the proposed algorithm. RMC is the coding information of the current status during encoding, and it is a favorable choice for developing a fast algorithm. The proposed algorithm comprises

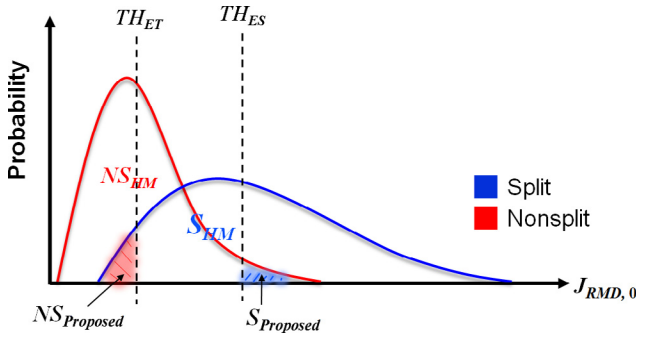


Fig. 2. Erroneous decisions due to ET/ES.

three parts: a CU partitioning strategy, PU intra mode decision, and TU depth decision. In the CU part, the RMC value is used to execute the ET or ES on the current CU. Regarding intra mode decisions, the ratio of the RMC of each mode over the least RMC mode is used to reduce the number of candidate modes. The TU partition of the mode with the least RMC is used to predict the TU partitions of the other modes.

3.1. Fast CU depth decisions

Although RMC is not equal to the RD cost of full RDO, it reveals the cost of coding a mode. If an intra mode is adequately predicted, its RMC should be low. Because the RD cost of a CU depends on the selected intra mode, we can determine whether the current CU depth is favorable by identifying the mode with the least RMC. Table 2 presents a summary of the simulation environment employed for gathering statistics in the algorithm development phase. To cover a variety of video characteristics, eight sequences were selected for training. The other settings were identical to the common test conditions [39] used in JCT-VC meetings.

Table 4
Percentages of split and nonsplit CUs (QP = 32).

Sequence	Decision	Depth 0	Depth 1	Depth 2
A_Traffic	Nonsplit	5.60%	32.83%	66.49%
	Split	94.40%	67.17%	33.51%
B_Cactus	Nonsplit	8.50%	34.33%	66.34%
	Split	91.50%	65.67%	33.66%
C_BasketballDrill	Nonsplit	0.44%	18.33%	53.45%
	Split	99.56%	81.67%	46.55%
E_FourPeople	Nonsplit	9.27%	37.09%	67.38%
	Split	90.73%	62.91%	32.62%

Table 5
Error rate selection criteria.

	Depth 0	Depth 1	Depth 2
ER_{ET} (%)	1–3	3–7	10–15
ER_{ES} (%)	10–15	3–7	1–3

After RMD, the selected N candidate modes for full RDO are sorted in ascending order according to RMC. The RMC value of each intra mode is denoted by $J_{RMD,\,i}$, where i is the index in the candidate list for full RDO, and the least RMC is $J_{RMD,\,0}$. Table 3 presents the statistics of the mean $J_{RMD,\,0}$ values for nonsplit and split CUs at the first three depths and two QPs, signifying that CUs with high RMC values are likely to be split because the prediction at the current depth is unsatisfactory. The $J_{RMD,\,0}$ values of nonsplit and split CUs are distinct, and $J_{RMD,\,0}$ varies with QPs and depths. Therefore, it is adopted for CU partitioning. In the proposed algorithm, the RMD procedure is performed before processing the current CU to obtain $J_{RMD,\,0}$, and the precalculated RMCs are stored for subsequent intra mode decisions; hence, no additional computation is required.

By comparing $J_{RMD,\,0}$ with a threshold, the proposed fast algorithm executes two CU partitioning strategies: ES and ET. A threshold TH_{ES} is set at each depth to execute ES on the current CU with high $J_{RMD,\,0}$. By contrast, a threshold TH_{ET} is set to conduct ET on the current CU with low $J_{RMD,\,0}$. If the $J_{RMD,\,0}$ value of a CU is lower than TH_{ET}, ET is performed; if it is higher than TH_{ES}, ES is performed. A range between two thresholds is reserved for "unsure" case, and in this range, making fast decisions is avoided to reduce prediction errors.

The selection of thresholds (TH_{ES}, TH_{ET}) is crucial for making CU depth decisions, and it entails balancing coding performance against saving time. In this work, the $J_{RMD,\,0}$ threshold in the training phase is selected by adjusting the error rate due to ET/ES. Subsequently, the selected thresholds are used for fitting models.

An erroneous decision is made in the execution of ET when originally split CUs (S_{HM}) are not split ($NS_{Proposed}$). By contrast, in the execution of ES, an erroneous decision is made when originally nonsplit CUs (NS_{HM}) are split ($S_{Proposed}$). The error rates engendered by ET and ES are denoted by ER_{ET} [Eq. (3)] and ER_{ES} [Eq. (4)], respectively. Fig. 2 illustrates the erroneous decisions induced by the proposed ET and ES.

$$ER_{ET}\ (\%) = \frac{NS_{Proposed}}{S_{HM}} \tag{3}$$

$$ER_{ES}\ (\%) = \frac{S_{Proposed}}{NS_{HM}} \tag{4}$$

Each threshold setting results in an error rate. Thus, selecting a suitable error rate enables us to determine the threshold. Accordingly, we gather statistics of the percentages of split and nonsplit CUs at each depth in the training sequences; Table 4 presents the results of four sequences. When most CUs are split, as is the case at depth 0 for each sequence, ES is preferable. Therefore, ER_{ES} can be higher (i.e., TH_{ES} shifts left in Fig. 2) to generate a greater number of split CUs. Simultaneously, ER_{ET} must be kept low to maintain RD performance because there are few nonsplit CUs. By contrast, CUs tend to not split at depth 2, and a higher ER_{ET} is tolerable; ER_{ES} must be kept low in this case. Briefly, a suitable error rate is selected on the basis of CU splitting/nonsplitting tendencies; Table 5 presents the selection criteria. The adaptive threshold for

Table 6
Curve-fitting parameters of thresholds for CU ES/ET.

	Depth	α_1	α_2	α_3	α_4	α_5	α_6	R^2
TH_{ES}	0	−31,820	−456.3	8606	16.2	33.97	−317.3	0.6376
	1	24,470	1296	−3747	−28.23	96.88	−70.83	0.9217
	2	2147	345.7	−300.7	−4.738	4.177	4.407	0.8917
TH_{ET}	0	5739	−788.8	2652	23.15	−116.3	202.8	0.8669
	1	4539	−253.2	607.8	2.759	2.202	−13.81	0.9651
	2	768.3	15.81	69.21	−0.4281	1.329	−1.253	0.8971

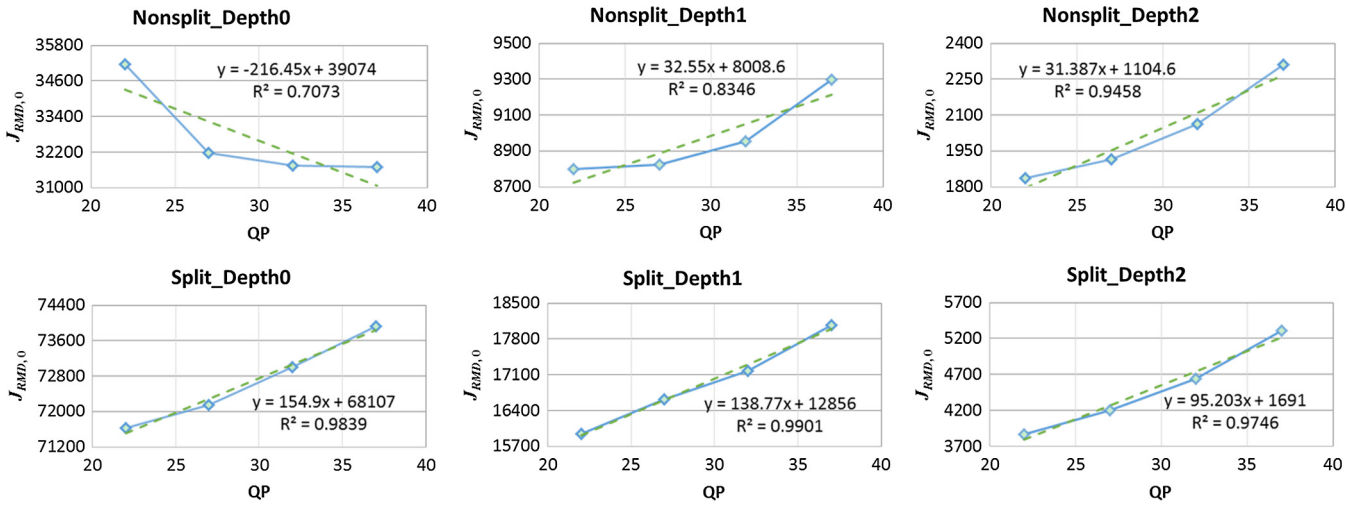


Fig. 3. Relationship between average $J_{RMD,\ 0}$ and QPs.

Table 7
Probability that the mode with the least RMC is the optimal.

Sequence	QP22	QP27	QP32	QP37	Average
A_Traffic	51.27%	59.73%	67.69%	76.01%	63.68%
B_Cactus	49.13%	63.09%	71.30%	79.30%	65.71%
C_BasketballDrill	60.44%	67.64%	72.64%	79.47%	70.05%
E_FourPeople	65.35%	71.97%	76.77%	81.94%	74.01%
Average	56.55%	65.61%	72.10%	79.18%	68.36%

fast decisions can then be modeled using the thresholds selected offline from training data. The training thresholds are further fine-tuned according to the generated encoding results.

Videos with different contents have different $J_{RMD,\ 0}$ values. Moreover, as shown in Table 3, $J_{RMD,\ 0}$ varies with QPs and CU depths. Hence, the proposed threshold model must adapt to both video content and QP at each depth. Notably, the thresholds are selected on the basis of the frame level and used for all CUs at each corresponding depth; accordingly, video characteristics are calculated at the frame level. To model the thresholds, two parameters, namely image complexity (C) and image gradient (G), are used to represent image properties in the proposed algorithm, and they are calculated as shown in Eqs. (5) and (6), respectively.

$$C = \frac{1}{W \times H}\sum_{i=0}^{W-1}\sum_{j=0}^{H-1}|Y_{i,j} - \mu_Y| \tag{5}$$

$$G = \frac{1}{(W-1)\times(H-1)}(G_{Hor} + G_{Ver})$$
$$G_{Hor} = \sum_{i=0}^{W-2}\sum_{j=0}^{H-2}|Y_{i,j} - Y_{i+1,j}|, G_{Ver} = \sum_{i=0}^{W-2}\sum_{j=0}^{H-2}|Y_{i,j} - Y_{i,j+1}| \tag{6}$$

where W and H are the width and height of the video, respectively; $Y_{i,j}$ is the pixel value at location (i,j); and μ_Y is the mean of all pixels in the current frame. To simplify the modeling process, the QP is first fixed at 32. Using the thresholds selected offline, the threshold at QP32, $TH(C, G, 32)$, is modeled by the quadric function expressed in Eq. (7). Six sets of parameters $(\alpha_1, \alpha_2, \ldots, \alpha_6)$ for TH_{ES} and TH_{ET} at each depth can be obtained by curve fitting. Table 6 presents the sets of fitting parameters as well as fitting accuracy in terms of R^2 values.

$$TH(C,G,32) = \alpha_1 + \alpha_2 \times C + \alpha_3 \times G + \alpha_4 \times C^2 + \alpha_5 \times C \times G + \alpha_6 \times G^2 \tag{7}$$

Because the parameters in Table 6 are applicable only to QP32, the threshold should be adapted to different QPs. The relationship between the average $J_{RMD,\ 0}$ value and QPs is then investigated. The relationship can be approximated by a linear function, and Fig. 3 presents the results; the dotted lines in this figure represent the estimated equations. The slope of the plot denotes the change in the average $J_{RMD,\ 0}$ value over QP, and it is incorporated into Eq. (7) to realize the final threshold model. The proposed adaptive thresholds can be calculated using Eqs. (8) and (9). The proposed algorithm calculates image property parameters at each frame level. The threshold is updated according to the current video content, and the proposed threshold model can adapt to image properties and QPs.

$$TH_{ES,Depth}(C,G,QP) = TH_{ES,Depth}(C,G,32) + slope_{ES,Depth} \times (QP - 32) \tag{8}$$

Table 8
Curve-fitting parameters of thresholds for the reduction of intra candidate modes.

	PU size	β_1	$\beta_2 \times 10^3$	$\beta_3 \times 10^2$	$\beta_4 \times 10^5$	$\beta_5 \times 10^4$	$\beta_6 \times 10^4$	R^2
TH_{mode}	64	1.24	3.023	−2.307	−4.244	1.563	2.03	0.9972
	32	1.235	2.361	−1.806	1.014	−1.735	6.861	0.9653
	16	1.245	5.073	−2.659	−7	0.4025	6.523	0.9247
	8	1.308	4.284	−2.891	0.7759	−2.843	11.52	0.9845
	4	1.366	−0.5522	−1.306	1.065	0.3671	2.15	0.9924

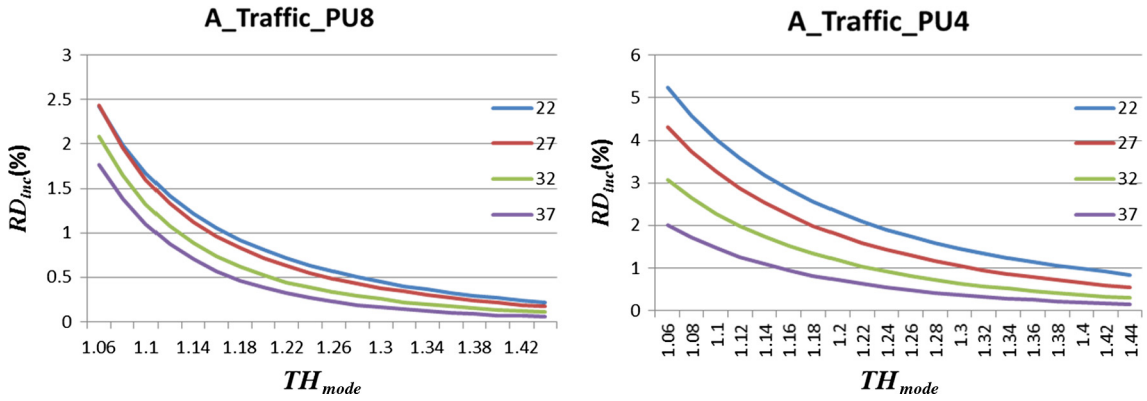


Fig. 4. Correlation between QP and TH_{mode} for different PUs.

$$TH_{ET,Depth}(C,G,QP) = TH_{ET,Depth}(C,G,32) + slope_{ET,Depth} \times (QP - 32) \tag{9}$$

3.2. Fast PU intra mode decisions

For fast intra mode decisions, we aim to reduce the number of modes in full RDO while maintaining the RMD process. In general, a prediction mode with a much lower RMC than those of other modes is likely to be the optimal mode. Table 7 presents the probability of finally selecting the mode with the least RMC as the optimal mode. The environmental settings are the same as those in Table 2. More than half of the modes with the least RMC are selected as the optimal modes, and the probability in Table 7 increases with the QP. This indicates that at higher QPs, the mode selected by RMD is closer to that selected by full RDO.

Because the mode with the least RMC is most likely to be the optimal one, it is always reserved for full RDO, and its cost ($J_{RMD,\,0}$) is used to determine the modes with high RMCs that should be removed. To compare the RMCs of modes, the RMC ratio $r_{RMC,i}$ is defined as shown in Eq. (10).

$$r_{RMC,i} = J_{RMD,i}/J_{RMD,0},$$
$$\text{where } i = \begin{cases} 1,2, & \text{for } 64\times64, 32\times32, 16\times16 \text{ PUs} \\ 1,2,3,4,5,6,7, & \text{for } 8\times8, 4\times4 \text{ PUs} \end{cases} \tag{10}$$

A high $r_{RMC,i}$ indicates that the RMC of the mode with index i is much higher than $J_{RMD,\,0}$. As previously mentioned, i is the index in the candidate list for full RDO rather than the directional index of the intra mode. A threshold TH_{mode} is set to determine whether a candidate mode has sufficiently high RMC to be removed from the candidate list. We propose removing the mode with index i if $r_{RMC,i} > TH_{mode}$. Notably, $r_{RMC,i} \leqslant r_{RMC,i+1}$ is always true because the modes in the candidate list are sorted in ascending order of cost after RMD. Moreover, the MPMs are removed if the RMC ratio of any mode exceeds TH_{mode}, because the subsequently included MPMs have higher RMCs than those of the original N modes selected by RMD. Therefore, if $r_{RMC,i} > TH_{mode}$, only the first i modes are reserved for full RDO.

To determine the suitable thresholds, a value called the RD increment (RD_{inc}) is calculated at the training stage, as defined in Eq. (11):

$$RD_{Inc.}\ (\%) = \frac{1}{K}\sum^{K}\frac{\min\{J_{FRD,i}\} - \min\{J_{FRD,j}\}}{\min\{J_{FRD,j}\}} \times 100 \tag{11}$$

where i is the number of remaining modes and K is the number of instances of candidate reduction when $r_{RMC,i} > TH_{mode}$ is satisfied. Moreover, j is the total number of intra modes in the candidate list (N + MPMs). RD_{inc} indicates that the average RD cost increases per fast decision made. To maintain a favorable trade-off between RD performance and time saving, the TH_{mode} value is selected such that RD_{inc} is less than 0.5%.

Similar to the method applied to obtain the thresholds for early CU splitting and termination, the two image-related parameters, complexity (C) and gradient (G), in Section 3.1 are used to model the threshold (TH_{mode}) selected offline. The threshold for fast intra mode decisions can be modeled through Eq. (12) with two video properties for each PU size at QP32. Table 8 lists the curve fitting results.

$$TH_{mode,PU\,size}(C,G,32) = \beta_1 + \beta_2\times C + \beta_3\times G + \beta_4\times C^2 + \beta_5 \times C\times G + \beta_6\times G^2 \tag{12}$$

The parameters in Table 8 are applicable only to QP32. The thresholds should vary with the QPs. For a fixed RD_{inc}, the threshold shows an approximately linear decrease with the QP, as Fig. 4 shows. This is consistent with the observation in Section 3.1. RMD demonstrates superior performance at higher QPs, and the threshold can be relaxed accordingly. Moreover, the threshold should vary to a greater degree in a smaller PU than in a larger PU. The rate of threshold variation is determined empirically, and the final model of TH_{mode} is given by Eq. (13). The number of intra modes for full RDO can be reduced by comparing the RMC ratio $r_{RMC,i}$ with the proposed TH_{mode}.

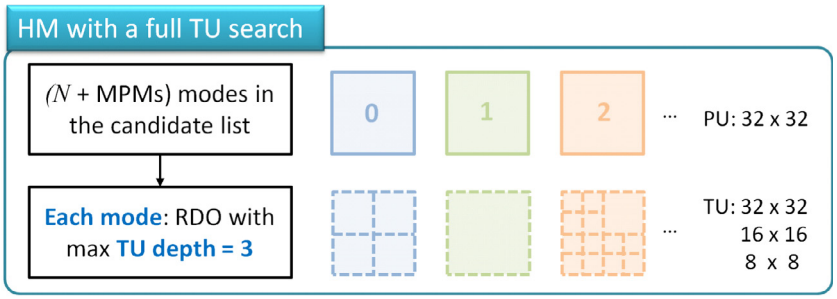
$$TH_{mode,PU\,size}(C,G,QP) = TH_{mode,PU\,size}(C,G,32) - d\times(QP-32), \tag{13}$$

where

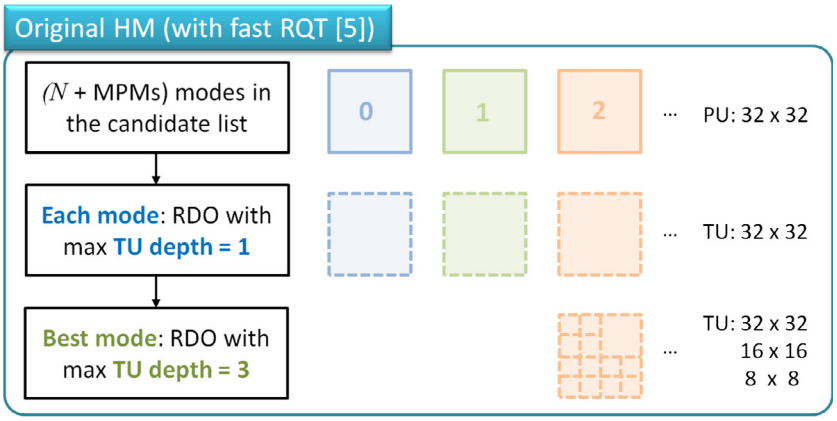
$$d = \begin{cases} 0.004 & for\ \ 64\times64,\ 32\times32, 16\times16 \text{ PUs} \\ 0.008 & for\ \ 8\times8 \text{ PU} \\ 0.012 & for\ \ 4\times4 \text{ PU} \end{cases}$$

3.3. Fast TU depth decisions

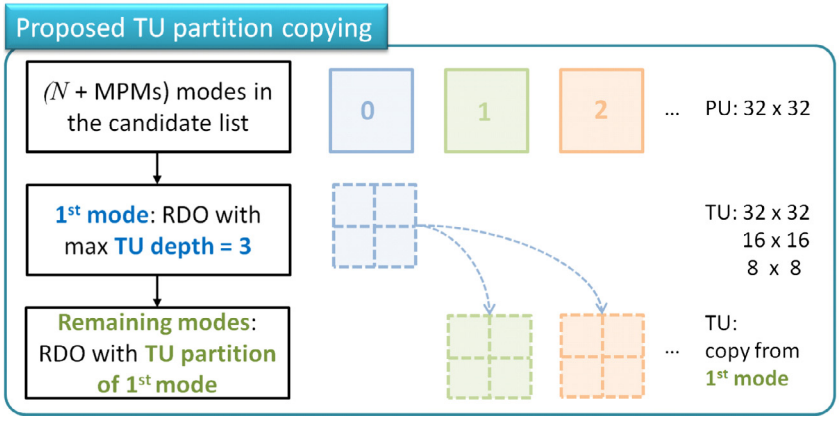
In general, an HEVC encoder should test all possible TU depths to determine the optimal TU partition of each intra mode. Fig. 5(a) illustrates an example of an HEVC full TU search. If the current PU size is 32×32, each candidate intra mode should examine all TU sizes (i.e., 32×32, 16×16, and 8×8 in this example) to determine the optimal TU partition. Subsequently, the RD cost of each intra mode (with its optimal TU partition) is compared to determine the optimal intra mode of this PU. The heuristic TU search can be simplified in the same manner as the fast algorithms proposed for the CU depth decision.



(a) HM with a full TU search.



(b) Original HM with a default fast RQT.



(c) Proposed fast TU depth decision.

Fig. 5. Illustrations of the TU depth decision.

However, few algorithms have been proposed specifically for the intra TU depth decision because a fast RQT method [5] for intra coding is built into reference software and enabled by default. Instead of testing all TU depths, the fast RQT algorithm first determines the optimal intra mode according to the maximum available TU size. Only the most favorable mode is examined using all TU depths to obtain the optimal TU partition. Fig. 5(b) depicts an example of the default fast RQT algorithm. Nevertheless, this default algorithm was developed on the basis of TMuC, a predecessor of HM, and the observed time saving was approximately 34% at that time. Table 9 presents the performance of the fast RQT algorithm for several training sequences in HM; five frames are encoded, and HM with a full TU search is used as the reference. The result is not as favorable as that on TMuC. A fast TU algorithm superior to the default one is attractive. In this paper, we propose a simple but efficient fast TU partitioning method that outperforms the default method.

Table 9
Performance of the default fast RQT in intra coding (HM with a full TU search is used as a reference).

Testing sequences	**HHI [5]**	
Class/Sequence	BDBR (%)	ΔT (%)
A_Traffic	0.701	−20.522
B_Cactus	0.707	−20.366
C_BasketballDrill	0.787	−20.188
E_FourPeople	0.909	−20.023
Average	0.776	−20.275

As shown in Table 7, for approximately 68% of intra mode decisions, the mode with the least RMC is selected as the final mode. In general, the differences in the prediction of various intra modes are limited for regions without highly complex textures. When the TU

Table 10
TU hit with the mode with the least RMC.

	CU64	CU32	CU16	CU8	Average
A_Traffic	82.18%	76.50%	75.29%	76.74%	77.68%
B_Cactus	82.12%	76.59%	76.14%	77.91%	78.19%
C_BasketballDrill	82.72%	73.52%	72.98%	73.61%	75.71%
E_FourPeople	81.43%	78.15%	79.79%	81.04%	80.10%
Average	82.11%	76.19%	76.05%	77.33%	77.92%

Table 11
Performance of the proposed fast TU depth decision (HM with a full TU search is used as a reference).

Testing sequences	Proposed TU Copy	
Class/Sequence	BDBR (%)	ΔT (%)
A_Traffic	0.577	−25.310
B_Cactus	0.484	−25.520
C_BasketballDrill	0.445	−25.157
E_FourPeople	0.721	−25.120
Average	0.557	−25.277

residual is similar, the TU is expected to adopt a similar partition. Therefore, we analyzed the probability that the TU partition of each intra mode in the candidate list for full RDO is the same as that of the mode with the least RMC. Table 10 presents the results, indicating that the average hit rate is nearly 78%. Thus, we propose storing the TU partition of the mode with the least RMC and simply copying this partition for the remaining modes to accelerate TU depth decisions. Fig. 5(c) shows the proposed method. The first mode (the mode with the least RMC) is tested with all possible TU sizes, and the TU partition of this mode is stored. The RD costs of the remaining intra modes are then calculated using the stored TU partition. Finally, the optimal intra mode is selected from all the candidate modes. Table 11 presents the performance of the proposed method, and HM with a full TU search serves as a reference. Compared with the original HM default fast RQT, the proposed strategy of copying the TU partition not only improves RD performance but also shortens the encoding time.

3.4. Summary of the overall algorithm

The overall fast algorithm proposed in this study is summarized in Fig. 6. All thresholds for fast decisions are calculated and updated at the frame level on the basis of the proposed models. The proposed fast CU decision algorithm is executed at CU depths of 0–2, and the RMD is processed early in these cases. The decision to split or terminate the current CU is made early by comparing the least RMC $J_{RMD,\ 0}$ with the proposed adaptive threshold. The number of intra modes for full RDO is lowered according to the RMC ratio $r_{RMC,\ i}$. The TU partition state of the mode with the least RMC is stored, and the TU partitions of the remaining modes are copied from it.

4. Experimental results

Table 12 presents a summary of the simulation environment; the other settings are the same as the common test conditions [39] used in JCT-VC meetings. The reference software program was executed in Microsoft Visual Studio 2013 on the x64 platform and in release mode. Coding efficiency was measured in terms of BD bit-rate (BDBR) (%) [41], and ΔT (%) represents the percentage of encoding time saved in comparison with the original HM 15.0, as presented in Eq. (14). The average PSNR defined in Eq. (15) was used for calculating BDBR. Four other algorithms [8,9,26,29] were also implemented on HM 15.0, and their performance levels were compared with that of the proposed algorithm. All schemes used the original HM as the reference.

$$\Delta T\ (\%) = \frac{1}{4}\sum_{i=1}^{QP_i}\frac{Enc.Time^{QP_i}_{proposed} - Enc.Time^{QP_i}_{HM15.0}}{Enc.Time^{QP_i}_{HM15.0}} \times 100,$$
$$QP_i = \{22, 27, 32, 37\} \tag{14}$$

$$PSNR_{AVG} = \frac{6PSNR_Y + PSNR_U + PSNR_V}{8} \tag{15}$$

4.1. Performance evaluation of the proposed algorithm

The performance of the proposed algorithm is summarized in Table 13. The proposed fast CU depth decision, fast PU intra mode decision, fast TU depth decision, and overall algorithms are denoted as Pro_CU, Pro_mode, Pro_TU, and Pro_overall, respectively. The built-in fast RQT algorithm for intra coding was disabled in the Pro_TU and Pro_overall cases.

The proposed fast CU decision reduced the encoding time by 38% on average with a 0.69% BDBR loss. Figs. 7 and 8 illustrate the CU partition results for the most and least favorable BDBR cases obtained from the proposed fast CU decision algorithm. The proposed algorithm tends to encode larger CUs in regions with homogeneous or simple texture (marked in red in Figs. 7 and 8) and smaller CUs in regions with complex texture (marked in green in Fig. 7), compared with the original HM. This tendency results in a limited effect on RD performance when the region is sufficiently homogeneous or complex, such as the marked parts in Fig. 7; hence, superior BDBR is achieved. In Fig. 8, the marked part is not complex but contains distinct lines or edges, and encoding large CUs in such regions degrades RD performance. The proposed algorithm is likely to make a wrong decision in the previously mentioned region because the $J_{RMD,\ 0}$ values of large and small CUs are not distinct.

For the proposed fast intra mode decision, the algorithm efficiently removes unnecessary modes, and the average RD loss is thus negligible. The proposed fast TU depth decision algorithm simultaneously improves RD performance and saves more time. By combining the proposed fast CU and intra mode decision algorithms, we were able to increase the reduction in encoding time from 38% to 48% with a slight increase in BDBR. The overall performance was enhanced further by the proposed TU partition copying strategy. Benefiting from the proposed fast TU depth decision algorithm, the average BDBR of the Pro_overall case was almost the same as that of Pro_CU. The overall fast algorithm reduced the encoding time at most by 59% and on average by 51%, compared with HM 15.0, with a BDBR loss of only 0.69%.

4.2. Performance of the proposed algorithm with/without consideration of QPs in threshold determination

This section demonstrates the effectiveness of QP consideration in determining thresholds for the proposed fast CU and PU intra mode algorithms. Table 14 compares performance with and without consideration of QPs. For both the fast CU depth decision and

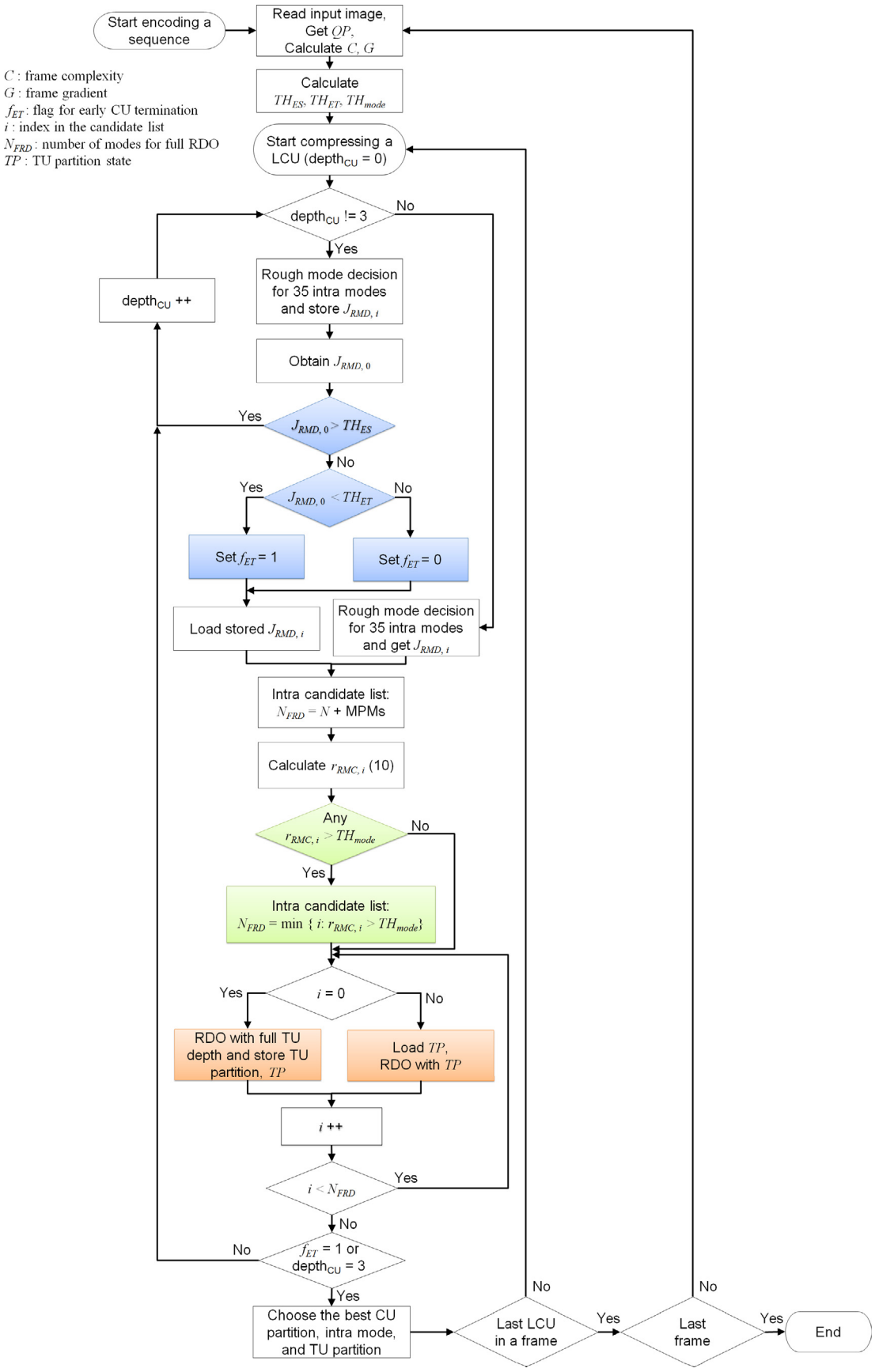


Fig. 6. Flowchart of the proposed overall algorithm.

Table 12
Simulation environment.

Reference software	HM 15.0 [40]	
Configuration	Intra_main	
Training sequences	ClassA_PeopleOnStreet	2560 × 1600 @ 30 fps
	ClassA_Traffic[a]	2560 × 1600 @ 30 fps
	CalssB_Kimono	1920 × 1080 @ 24 fps
	CalssB_ParkScene[a]	1920 × 1080 @ 24 fps
	ClassB_BasketballDrive	1920 × 1080 @ 50 fps
	ClassB_BQTerrace	1920 × 1080 @ 60 fps
	ClassB_Cactus[a]	1920 × 1080 @ 50 fps
	ClassC_BasketballDrill[a]	832 × 480 @ 50 fps
	ClassC_BQMall[a]	832 × 480 @ 60 fps
	ClassC_PartyScene	832 × 480 @ 50 fps
	ClassC_RaceHorsesC	832 × 480 @ 30 fps
	ClassD_BasketballPass	416 × 240 @ 50 fps
	ClassD_BlowingBubbles	416 × 240 @ 50 fps
	ClassD_BQSquare[a]	416 × 240 @ 60 fps
	ClassD_RaceHorse[a]	416 × 240 @ 30 fps
	ClassE_FourPeople[a]	1280 × 720 @ 60 fps
	ClassE_Johnny	1280 × 720 @ 60 fps
	ClassE_KristenAndSara	1280 × 720 @ 60 fps
QP	22, 27, 32, 37	
FramesToBeEncoded	30	
Hardware		
CPU	Intel(R) Core(TM) i7-2600 @ 3.40 GHz	
RAM	4.0 G bytes	

[a] Sequences used for training.

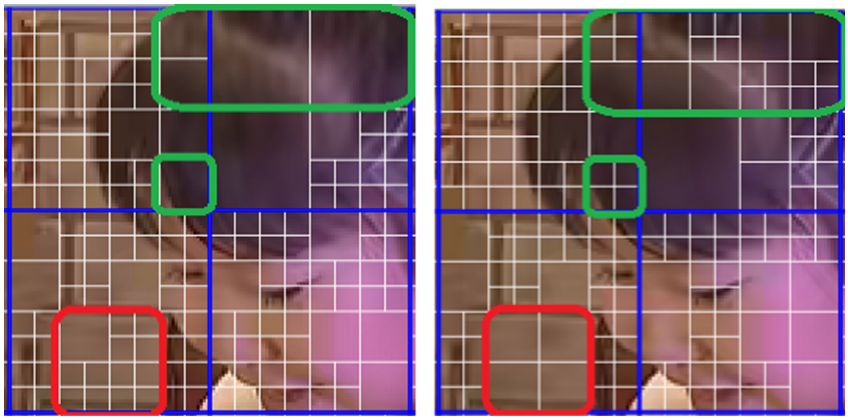
fast intra mode decision, the average RD performance and time saving are improved by considering QPs in threshold determination. Because the features used for fast decisions vary with QPs, incorporating QPs into threshold determination to adapt to variation can improve performance.

4.3. Performance comparison with existing algorithms

Table 15 shows the simulation results of existing fast intra coding algorithms [8,9,26,29]. The variance of the current CU was compared with a fixed threshold for early CU termination and splitting in [9]; thus, the algorithm can perform adequately only in select sequences. Making decisions without reserving the "unsure" case saves a considerable amount of time but substantially degrades RD performance. Shen et al. calculated the texture homogeneity of CUs for early CU termination and used the neighboring CU depth information for early CU splitting [8]. This resulted in superior BDBR to that in [9], but less time was saved. Kim et al. proposed a hierarchical mode decision algorithm to reduce candidate modes for RMD and full RDO [26]. A RD cost prediction model was established to replace the RMD process, and the number of modes for full RDO was reduced for 8 × 8 and 4 × 4 PUs [29]. Both [26] and [29] applied a fixed and reduced number of modes for full RDO; hence, a stable time-saving performance was achieved for

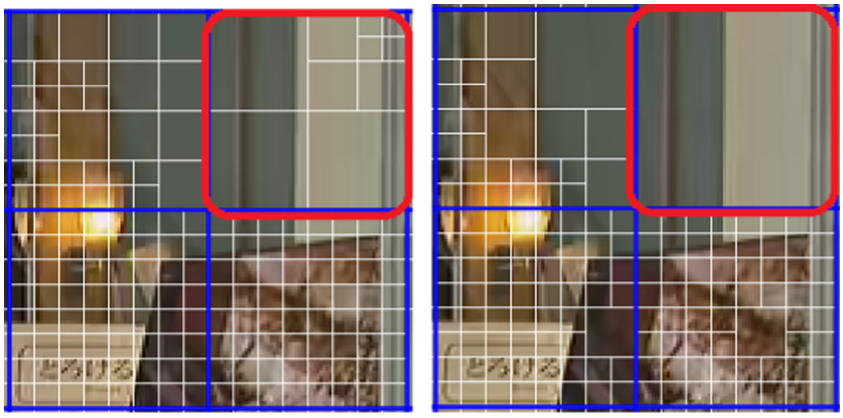
Table 13
BDBR and time-saving performance of the proposed algorithm.

Sequences	Pro_CU		Pro_mode		Pro_TU		Pro_CU + mode		Pro_overall	
Class/Sequence	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
A_PeopleOnStreet	0.70	−30.05	0.28	−16.29	−0.21	−2.72	1.05	−42.09	0.87	−45.77
A_Traffic	0.46	−31.55	0.19	−15.42	−0.13	−4.49	0.67	−41.31	0.54	−44.72
B_Kimono	0.32	−32.61	0.04	−12.61	0.15	−5.36	0.32	−38.59	0.49	−43.32
B_ParkScene	0.38	−31.36	0.12	−11.61	−0.09	−4.71	0.55	−38.70	0.44	−42.04
B_BasketballDrive	0.30	−48.76	0.03	−14.25	0.01	−5.51	0.30	−54.48	0.40	−57.57
B_BQTerrace	1.17	−43.62	0.15	−20.76	−0.27	−4.09	1.39	−55.16	1.14	−58.56
B_Cactus	0.34	−29.88	0.12	−12.93	−0.19	−4.64	0.49	−38.27	0.33	−42.21
C_BasketballDrill	0.59	−35.19	0.09	−17.64	−0.35	−4.38	0.67	−47.62	0.40	−50.89
C_BQMall	1.30	−38.86	0.27	−19.29	−0.32	−3.60	1.55	−51.37	1.21	−54.39
C_PartyScene	0.32	−33.75	0.27	−17.15	−0.20	−2.62	0.59	−45.59	0.35	−48.22
C_RaceHorsesC	1.10	−37.12	0.11	−15.10	−0.23	−4.04	1.23	−46.87	1.05	−50.04
D_BasketballPass	1.24	−45.34	0.09	−17.68	−0.29	−3.62	1.46	−56.09	1.17	−58.95
D_BlowingBubbles	0.28	−29.87	0.16	−16.14	−0.31	−2.41	0.47	−42.87	0.25	−45.86
D_BQSquare	0.74	−42.75	0.32	−14.90	−0.28	−2.06	1.02	−53.78	0.62	−56.04
D_RaceHorses	0.62	−32.05	0.31	−17.31	−0.20	−2.97	0.93	−45.09	0.69	−48.51
E_FourPeople	0.86	−40.47	0.24	−17.30	−0.23	−3.84	1.04	−49.95	0.86	−53.09
E_Johnny	0.84	−47.30	0.15	−15.51	−0.22	−4.14	0.88	−54.58	0.74	−57.58
E_KristenAndSara	0.92	−48.65	0.26	−14.83	−0.26	−4.77	1.13	−55.48	0.86	−58.43
Average	0.69	−37.73	0.18	−15.93	−0.20	−3.89	0.87	−47.66	0.69	−50.90



(a) HM 15.0 (b) Proposed fast CU decision

Fig. 7. CU partition results of a part of D_BlowingBubbles at QP32 (2nd frame). The blue line is the LCU boundary. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) HM 15.0 (b) Proposed fast CU decision

Fig. 8. CU partition results of a part of C_BQMall at QP32 (2nd frame). The blue line is the LCU boundary. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 14
Performance comparison of the proposed fast CU and intra mode algorithms with/without consideration of QPs.

Sequences	Pro_CU		Pro_CU_without QP		Pro_mode		Pro_mode_without QP	
Class	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
A	0.58	−30.80	0.63	−29.83	0.24	−15.86	0.29	−13.43
B	0.50	−37.25	0.53	−36.68	0.09	−14.43	0.13	−13.02
C	0.83	−36.23	0.93	−34.53	0.19	−17.30	0.25	−17.46
D	0.72	−37.50	0.81	−35.83	0.22	−16.51	0.29	−16.74
E	0.87	−45.47	0.88	−44.34	0.21	−15.88	0.26	−15.50
Average	0.69	−37.73	0.75	−36.53	0.18	−15.93	0.23	−15.29

Table 15
Comparison of the BDBR and time-saving performance of the proposed algorithm with those of the existing algorithms.

Sequences	Nishikori [9]		Shen [8]		Kim [26]		Tariq [29]		Pro_overall	
Class/Sequence	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
A_PeopleOnStreet	4.12	−54.09	0.79	−35.55	1.24	−32.79	1.06	−23.59	0.87	−45.77
A_Traffic	4.99	−57.86	1.14	−38.55	1.10	−32.58	0.92	−23.36	0.54	−44.72
B_Kimono	18.87	−69.11	0.61	−29.63	0.44	−33.74	0.50	−24.29	0.49	−43.32
B_ParkScene	4.23	−60.08	0.77	−37.90	0.49	−33.37	0.73	−23.47	0.44	−42.04
B_BasketballDrive	9.24	−71.83	1.99	−60.46	0.73	−33.71	0.73	−24.27	0.40	−57.57
B_BQTerrace	2.30	−55.09	0.61	−38.24	1.05	−33.94	0.59	−25.11	1.14	−58.56
B_Cactus	5.05	−59.01	1.30	−38.27	1.07	−32.72	0.77	−23.45	0.33	−42.21
C_BasketballDrill	5.32	−62.25	0.52	−29.68	1.83	−31.54	0.81	−23.23	0.40	−50.89
C_BQMall	2.82	−49.33	0.84	−34.86	1.31	−32.11	0.98	−23.33	1.21	−54.39
C_PartyScene	0.63	−39.90	0.08	−22.24	1.44	−31.49	0.92	−22.46	0.35	−48.22
C_RaceHorsesC	2.73	−50.94	0.46	−29.83	0.97	−31.72	0.73	−22.69	1.05	−50.04
D_BasketballPass	3.95	−59.44	0.78	−32.94	1.37	−32.25	0.87	−23.88	1.17	−58.95
D_BlowingBubbles	1.43	−43.45	−0.02	−17.13	1.65	−31.76	0.94	−23.31	0.25	−45.86
D_BQSquare	0.60	−46.13	0.13	−22.08	2.11	−31.72	0.98	−22.43	0.62	−56.04
D_RaceHorses	2.08	−43.82	0.08	−19.52	1.58	−31.75	1.09	−22.99	0.69	−48.51
E_FourPeople	5.27	−59.02	2.00	−47.52	1.16	−33.34	1.05	−24.03	0.86	−53.09
E_Johnny	8.77	−71.36	3.74	−56.09	1.28	−33.92	1.05	−24.54	0.74	−57.58
E_KristenAndSara	6.39	−68.48	3.77	−56.75	1.41	−33.16	0.90	−24.05	0.86	−58.43
Average	4.93	−56.73	1.09	−35.96	1.23	−32.65	0.87	−23.58	0.69	−50.90

each sequence. Compared with the existing algorithms, the proposed algorithm achieves a superior trade-off between RD performance and time saving.

5. Conclusions

We propose an RMC-based fast intra coding algorithm for HEVC. The originally generated information (RMC) during intra coding is used to achieve the fast CU depth decision, fast intra mode decision, and fast TU depth decision. The least RMC in the current CU is used for early CU splitting and termination because the RMCs of split CUs often differ from those of nonsplit CUs. The adaptive threshold for fast CU decisions is modeled by a function of two image properties (complexity and gradient) and the QP value. Intra modes with higher RMCs have a lower probability of being the optimal mode than do those with lower RMCs. Hence, the RMC ratio, described as the ratio between the least RMC and the RMC values of all other candidate modes, is used to reduce the number of modes for full RDO. The fast decision threshold is also obtained

by fitting offline data, in addition to modeling it by a function of the aforementioned two image properties. For fast TU depth decisions, we propose a TU partition copying scheme based on the TU partition of the mode with the least RMC. The scheme is simple but efficient, and we suggest that it should replace the built-in fast RQT method in reference software to enhance coding efficiency. The proposed algorithm reduces encoding time by approximately 51% with a BDBR loss of only 0.69%, and it is competitive with state-of-the-art algorithms.

References

[1] G.J. 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) (2012) 1649–1668.

[2] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 560–576.

[3] J. Lainema, F. Bossen, W.J. Han, J.H. Min, Intra coding of the HEVC standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1792–1801.

[4] L. Zhao, L. Zhang, X. Zhao, S. Ma, D. Zhao, W. Gao, Further encoder improvement of intra mode decision, JCTVC-D283, 4th JCT-VC meeting, Daegu, KR, January 2011, pp. 1–4.

[5] B. Bross, H. Kirchhoffer, H. Schwarz, T. Wiegand, Fast intra encoding for fixed maximum depth of transform quadtree, JCTVC-C311, 3rd JCT-VC meeting, Guangzhou, CN, October 2010, pp. 1–6.

[6] C. Bai, C. Yuan, Fast coding tree unit decision for HEVC intra coding, IEEE ICCE-China Workshop, Shenzhen, China, April 2013, pp. 28–31.

[7] J.W. Qiu, F. Liang, Y.L. Luo, A fast coding unit selection algorithm for HEVC, in: IEEE International Conference on Multimedia and Expo Workshops (ICMEW), California, USA, July 2013, pp. 1–5.

[8] L. Shen, Z. Zhang, Z. Liu, Effective CU Size Decision for HEVC Intracoding, IEEE Trans Image Process 23 (10) (2014) 4232–4241.

[9] T. Nishikori, T. Nakamura, T. Yoshitome, K. Mishiba, A fast CU decision using image variance in HEVC intra coding", in: Proc. IEEE Symposium on Industrial Electronics and Applications (ISIEA), Kuching, Malaysia, September 2013, pp. 52–56.

[10] Q. Zhang, J. Sun, Y. Duan, Z. Guo, A two-stage fast CU size decision method for HEVC intracoding, in: Proc. IEEE 17th International Workshop on Multimedia Signal Processing (MMSP), October 2015, pp. 1–6.

[11] M.U.K. Khan, M. Shafique, J. Henkel, An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding, in: Proc. IEEE International Conference on Image Processing (ICIP), Melbourne, VIC, September 2013, pp. 1578–1582.

[12] W. Shi, X. Jiang, T. Song, T. Shimamoto, Edge information based fast selection algorithm for intra prediction of HEVC, in: Proc. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Ishigaki, November 2014, pp. 17–20.

[13] B. Min, Ray C.C. Cheung, A fast CU size decision algorithm for the HEVC intra encoder, IEEE Trans Circuits Syst Video Technol 25 (5) (2015) 892–896.

[14] T. Mallikarachchi, A. Fernando, H.K. Arachchi, Efficient coding unit size selection based on texture analysis for HEVC intra prediction, in: Proc. IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, July 2014, pp. 1–6.

[15] M. Radosavljevíc, G. Georgakarakos, S. Lafond, D. Vukobratovíc, Fast coding unit selection based on local texture characteristics for HEVC intra frame, in: Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP), December 2015, pp. 1377–1381.

[16] W. Geuder, P. Amon, E. Steinbach, Low-complexity block size decision for HEVC intra coding using binary image feature descriptors, in: Proc. IEEE International Conference on Image Processing (ICIP), Quebec City, QC, September 2015, pp. 242–246.

[17] L. Shen, Z. Zhang, P. An, Fast CU size decision and mode decision algorithm for HEVC intra coding, IEEE Trans. Consum. Electron. 59 (1) (2013) 207–213.

[18] G. Kim, C. Yim, Adaptive CU splitting and pruning method for HEVC intra coding, Electron. Lett. 50 (10) (2014) 748–750.

[19] X. Shang, G. Wang, T. Fan, Y. Li, Fast CU size decision and PU mode decision algorithm in HEVC intra coding, in: Proc. IEEE International Conference on Image Processing (ICIP), Quebec City, QC, September 2015, pp. 1593–1597.

[20] 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) (2013) 1555–1564.

[21] K. Lim, J. Lee, S. Kim, S. Lee, Fast PU skip and split termination algorithm for HEVC intra prediction, IEEE Trans. Circuits Syst. Video Technol. 25 (8) (2015) 1335–1346.

[22] Y. Zhang, S. Kwong, G. Zhang, Z. Pan, H. Yuan, G. Jiang, Low complexity HEVC intra coding for high-quality mobile video communication, IEEE Trans. Industrial Inform. 11 (6) (2015) 1492–1504.

[23] C.F. Tseng, Y.T. Lai, Fast coding unit decision and mode selection for intra-frame coding in high-efficiency video coding, IET Image Proc. 10 (3) (2016) 215–221.

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

[25] D. Palomino, E. Cavichioli, A. Susin, L. Agostini, M. Shafique, J. Henkel, Fast HEVC intra mode decision algorithm based on new evaluation order in the coding tree block, in: Proc. Picture Coding Symposium (PCS), San Jose, CA, December 2013, pp. 209–212.

[26] T.S. Kim, M.H. Sunwoo, J.G. Chung, Hierarchical fast mode decision algorithm for intra prediction in HEVC, in: Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, May 2015, pp. 2792–2795.

[27] L. Gao, S. Dong, W. Wang, R. Wang, W. Gao, Fast intra mode decision algorithm based on refinement in HEVC", in: Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, May 2015, pp. 517–520.

[28] S. Wang, S. Ma, X. Jiang, J. Fan, D. Zhao, W. Gao, A fast intra optimization algorithm for HEVC, in: Proc. IEEE Visual Communications and Image Processing Conference, Valletta, December 2014, pp. 241–244.

[29] J. Tariq, S. Kwong, H. Yuan, HEVC intra mode selection based on rate distortion (RD) cost and sum of absolute difference (SAD), J. Vis. Commun. Image Represent. 35 (February) (2016) 112–119.

[30] W. Li, R. Wang, J. Wang, D. Xu, J. Xu, A fast mode decision algorithm for intra prediction in HEVC, in: Proc. International Conference on Computer Science and Service System (CSSS), 2014, pp. 587–590.

[31] K. Choi, E.S. Jang, Early TU decision method for fast video encoding in high efficiency video coding, Electron. Lett. 48 (12) (2012) 689–691.

[32] Y. Shi, Z. Gao, X. Zhang, Early TU split termination in HEVC based on quasi-zero-block, in: Proc. 3rd International Conference on Electric and Electronics (EEIC), Hong Kong, China, November 2013, pp. 450–454.

[33] J. Kang, H. Choi, J.G. Kim, Fast transform unit decision for HEVC, in: Proc. IEEE International Congress on Image and Signal Processing (CISP), Hangzhou, China, December 2013, pp. 26–30.

[34] Z. Lv, S. Dong, R. Wang, X. Xie, H. Jia, W. Wang, W. Gao, An all-zero blocks early detection method for high efficiency video coding, in: Proc. SPIE, vol. 9029, February 2014, pp. 902902–1–902902-7.

[35] P.T. Chiang, T.S. Chang, Fast zero block detection and early CU termination for HEVC video coding, in: Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, May 2013, pp. 1640–1643.

[36] S.W. Teng, H.. Hang, Y.F. Chen, Fast mode decision algorithm for residual quadtree coding in HEVC, in: Proc. Visual Communications and Image Processing (VCIP), November 2011, pp. 1–4.

[37] Y. Zhang, Z. Li, M. Zhao, B. Li, Fast residual quad-tree coding for the emerging high efficiency video coding standard, China Commun. 10 (10) (2013) 155–166.

[38] C.C. Wang, Y.C. Liao, J.W. Wang, C.W. Tung, An effective TU size decision method for fast HEVC encoders, in: Proc. International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, June 2014, pp. 1195–1198.

[39] F. Bossen, Common test conditions and software reference configurations, JCTVC-L1100, in: 12th JCT-VC meeting, Geneva, CH, January 2013, pp. 1-4.

[40] I.K. Kim, K. McCann, K. Sugimoto, B. Bross, W.J Han, G.J. Sullivan, High efficiency video coding (HEVC) test model 15 (HM15) encoder description, JCTVC-Q1002, IN: 17th JCT-VC meeting, Valencia, ES, April 2014, pp. 1-37.

[41] G. Bjontegaard, Calculation of average PSNR differences between RD-curves, VCEG-M33, in: ITU-T Q.6/SG16 VCEG 13th meeting, Austin, Texas, USA, April 2001, pp. 1–5.