Multimed Tools Appl (2017) 76:2001–2017
DOI 10.1007/s11042-015-3155-7

Fast CU size decision and mode decision algorithm for intra prediction in HEVC

Yun Song [1,2,3] · Ye Zeng [1,3] · Xueyu Li [1,3] · Biye Cai [1,3] · Gaobo Yang [2]

Received: 7 June 2015 / Revised: 7 December 2015 / Accepted: 9 December 2015 /
Published online: 6 January 2016

Abstract In the intra prediction process, High Efficiency Video Coding (HEVC) provides a quadtree-based coding unit (CU) block partitioning structure and up to 35 kinds of prediction modes to improve the coding performance. These technologies improve the coding efficiency significantly while the coding complexity is simultaneously increased rapidly as well. In this paper, a novel fast CU size decision and mode decision algorithm is proposed for the intra prediction of HEVC. The overall algorithm consists of two processes, the fast CU size decision and fast mode decision. In the fast CU size decision process, we adopt an adaptive discretization total variation (DTV) threshold-based CU size determination algorithm to skip some specific depth levels. In the fast mode decision process, an orientation gradient-based mode decision is proposed to reduce the candidate modes involved in the rough mode decision (RMD) and the rate distortion optimization (RDO) process. The experimental results on the HEVC reference software HM demonstrate that the proposed algorithm can significantly reduce the coding time with negligible performance loss.

✉ Yun Song
sonie@126.com

Ye Zeng
980193148@qq.com

Xueyu Li
401756139@qq.com

Biye Cai
512932680@qq.com

Gaobo Yang
yanggaobo@hnu.edu.cn

[1] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha, China

[2] School of Information Science & Engineering, Hunan University, Changsha, China

[3] Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science & Technology, Changsha, China

Keywords High efficiency video coding · CU size decision · Mode decision algorithm · Intra prediction

1 Introduction

High Efficiency Video Coding (HEVC) [16] developed by the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, is the current state-of-the-art video coding standard. It aims to save 50 % rate cost while maintaining the same video quality over the predecessor H.264/AVC [18]. HEVC introduces many coding tools to improve the coding efficiency, such as flexible coding structures including the coding unit (CU), prediction unit (PU), transform unit (TU), quadtree-based CU block partitioning [7], and up to 35 intra prediction modes [17], etc. These tools achieve considerable coding performance but accompanied with much higher computational complexity which restricts its potential use in real-time applications. It is necessary to reduce the encoding complexity of HEVC with some acceptable performance loss for real-time applications.

HEVC adopts recursive quadtree coding units and rate distortion optimization (RDO) to traverse 35 intra prediction modes (including 33 directional prediction modes, DC mode, and planar (PLANAR) mode) to find the best mode to encode. On one hand, the more flexible intra prediction improves the coding efficiency by about 22 % on average compared with H.264 [8]. On the other hand, since HEVC supports many more block sizes and more prediction modes than H.264/AVC, it leads to much higher computational complexity. For the purpose of reducing coding time, many fast CU size decision and mode decision algorithms have been proposed. For fast CU size decision, Kim et al. [6] introduced an early termination method based on the statistics of rate-distortion costs in the CU splitting process. This method can save 24 % encoding time on average with only a negligible loss of coding efficiency with respect to the official HEVC reference model HM [4]. However, the thresholds employed for rate-distortion (RD) cost computing were obtained by training from a large amount of videos offline. Shen et al. [12] determined the CU depth range (including the minimum depth level and the maximum depth level) and skipped some specific depth levels rarely used in the previous frame and the neighboring CUs. Cho et al. [2] proposed a fast CU splitting and pruning method following a Bayesian decision rule based on low-complexity RD costs and full RD costs. Since the statistical parameters for the early CU splitting and pruning processes were periodically updated for each CU depth in response to changes of video characteristics, this method saved almost 50 % encoding time. Li et al. [9] speeded the process of partition for CTU by a fast CU splitting and pruning method, in which CUs are classified by their sizes, RD costs and Hadamard costs. Experiments reported this method reduced 46 % computational complexity on average with 0.82 % Bjontegaard delta rate (BD-rate) increase. For fast mode decision, Jiang et al. [5] presented a gradient-based fast mode decision algorithm by calculating the gradient for each direction and generating the gradient-mode histogram for each coding unit, which reduced the candidate modes involved in the rough mode decision (RMD) and RDO processes. This algorithm achieved almost 20 % encoding time reduction with only 0.74 % BD-rate increase. Yan et al. [19] presented an early termination and pixel-based edge detection method by reducing the number of candidates for the RDO process heuristically. It was reported that the algorithm saved 23.52 % encoding time with a negligible performance loss compared to HM 7.0. By analyzing the relation between a block's texture characteristics and its best coding mode, Zhang et al. [21] proposed an adaptive strategy for the fast mode decision in the intra coding of HEVC. Experimental results showed this method saved about 15 % and 20 % encoding time while BD-rate increments are only 0.64 % and 1.05 % in the all intra HE and all intra LC test condition on HM 4.0. Fini et al. [3] introduced a two stage algorithm for fast intra mode decision in HEVC, where in the first stage the number of the tested modes in RMD is reduced from 35 to 19. And in the second stage, the number of the tested modes in the RDO process is reduced as well.

The aforementioned algorithms have achieved considerable encoding time saving with little loss of coding efficiency. However, most of these algorithms consider the fast CU size determination or the fast mode decision solely. Actually, in the intra prediction process of HEVC, CU sizes are divided recursively from largest coding unit (LCU) to smallest coding unit (SCU). In each recursion, the current PU traverses 35 kinds of prediction modes for finding the optimal prediction mode. If only the fast CU size decision is considered, then 35 prediction modes must be computed for a certain size PU. Similarly, if only considering the fast mode decision, all the CU sizes from LCU to SCU are required to be calculated as well. Thus, the computational complexity with either one of these two types is still very high. Conversely, if the effective fast CU size decision and mode decision are simultaneously considered in the intra prediction process, it is possible to reduce the coding time dramatically with little encoding performance loss. In fact, some researchers have proposed a few fast algorithms which combine the fast CU size decision and fast mode decision. Zhang et al. [20] proposed a Hadamard cost model based on the progressive rough mode search to selectively check the potential modes instead of traversing all candidates in the fast mode decision. And, CU splitting is early terminated also if the estimated R-D cost is already larger than the R-D cost of the current CU. The experiments demonstrated the proposed algorithm achieves the state-of-art performance. Shen et al. [13] fully exploited the correlations between parent CUs in the upper depth levels or spatially nearby CUs and the current CU, and proposed a fast CU size decision and mode decision algorithm for HEVC intra coding. However, most of aforementioned methods only exploited the spatial or temporal correlations exist in digital videos for the fast CU splitting, whereas the relationship between video contents and optimal CU depth levels was not fully studied. Moreover, most of the existing fast mode decision algorithms filtered modes only by the texture direction and intensity detected in the current CU, which did not take full advantage of the intra-prediction mechanism characteristic in HEVC. To alleviate these drawbacks, this paper presents a fast and comprehensive method for both fast CU splitting and fast mode decision in the HEVC intra coding.

In fact, the optimal CU depth level is highly correlated with video texture characteristics. Larger CU sizes commonly do not result in a desirable RD performance for texture regions, whereas for plat regions, smaller CU sizes likely do not achieve the best coding performance. Based on this observation, we employ the discretization total variation (DTV) [15] to identify the complexity of a CU and present an adaptive DTV threshold-based method to filter or early terminate some specific depth levels for fast CU splitting. Furthermore, following the intra prediction mechanism characteristic in HEVC, an orientation gradient-based mode decision is proposed to further reduce the modes involved in the RMD and RDO processes for the fast mode decision. Since some unlikely CUs and directional modes are skipped in the intra prediction process in two stages, it is desirable to significantly reduce the computational complexity. Meanwhile we just skip the CUs and directional modes which likely do not lead to the desirable coding performance for the current CU, thus the coding efficiency is also maintained well. Experiments illustrate the advantage of the proposed algorithm. It exceeds the current state-of-art methods in terms of the coding time reduction and coding efficiency loss.

The remainder of this paper is organized as follows. Section 2 introduces the intra prediction framework used in HEVC. Then, Section 3 describes the proposed fast CU size decision and mode decision algorithm for intra prediction in HEVC. Experimental results are given in Section 4. Finally, we conclude this paper in Section 5.

2 Preliminaries of the intra prediction in HEVC

Compared with H.264/AVC, HEVC provides a more flexible and efficient encoding structure by introducing three block concepts: CU, PU and TU. These concepts replace the macroblock (MB) in H.264/AVC. CU is the basic unit of video coding which must be a square shaped block. A CU can be coded directly or be split to a set of smaller CUs. The maximum size of a CU is 64 × 64 (i.e., Largest Coding Unit, LCU) and the minimum size is 8 × 8 (i.e., Smallest Coding Unit, SCU). A CU can be regarded as one PU or divided to more PUs to implement the prediction process. In HEVC intra coding, there are only two partitions of the CU are allowed. That is, a CU is regarded as a whole PU or divided to 4 same-sized PUs to predict. TU is the elementary unit for transforming and quantizing whose size can range from 4 × 4 to 32 × 32. As illustrated in Fig. 1, HEVC adopts the quadtree-based CU partitioning structure during the intra prediction process. Specifically, each CU recursively splits into four equally sized blocks starting from LCU to SCU, and the SCU supports PU types of $2N\times2N$ and $N\times N$ partitions. Here, $2N\times2N$ is the size of the CU, and N is the half of the width or height of the CU.

For each PU, HEVC provides up to 35 intra prediction modes, as shown in Fig. 2 including 33 directional modes, DC mode and PLANAR mode, to select the best mode.

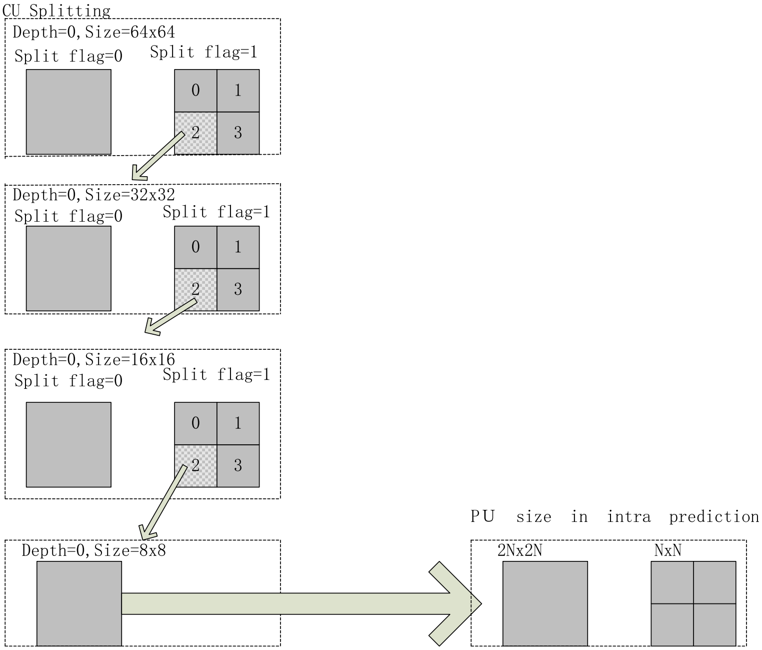


Fig. 1 Illustration of recursive CU partitioning structure and PU types in intra prediction for HEVC

In order to alleviate the computational complexity, the RMD process is first done to determine some best candidate modes. Specifically, the minimum absolute sums of Hadamard transformed coefficients of the residual signal and the mode bits for all the 35 modes are calculated as the simplified RD costs firstly [11]. A set of modes with smaller simplified RD costs are picked as candidate modes. The numbers of the candidate modes for 64 × 64, 32 × 32, 16 × 16, 8 × 8 and 4 × 4 sized PUs are 3, 3, 3, 8 and 8, respectively. Then, the most probable modes (MPM) are also added into the candidate mode set to perform RDO [22].

As described above, for LCU, a total of $(4^0 + 4^1 + 4^2 + 4^3 + 4^4)$ =341 different PU blocks are needed to be traversed for selecting the best CU splitting. Moreover, $341 \times 35 = 11935$ times simplified R-D costs are executed in the RMD and more than $(4^0 \times 3 + 4^1 \times 3 + 4^2 \times 3 + 4^3 \times 8 + 4^4 \times 8)$ =2623 times full R-D costs are computed in the RDO. Therefore, the intra prediction process is very time-consuming.

3 Proposed fast intra prediction algorithm

3.1 Adaptive discretization total variation threshold-based CU size determination

HEVC executes all depth levels ranging from 0 to 3, so the PU size scopes are from 64 × 64 to 4 × 4. However, the best CU partition has a strong correlation with the texture features of current treeblock. In practice, homogeneous areas tend to larger block sizes, whereas regions with rich textures use smaller block sizes more often. Figure 3 shows the CU splitting result of the first frame of the BasketballDrill sequence picked from [1] for testing using only intra prediction. As seen in Fig. 3, regions with complex textures tend to 8 × 8 or 4 × 4 PU sizes, whereas homogeneous regions are more likely to 16 × 16 or 32 × 32 sizes. That is, the best CU size can be estimated by the complexity of the CU. In this paper, we adopt the discretization total variation (DTV) [14] to measure the complexity of a CU. And then, an adaptive DTV threshold-based CU size determination algorithm is presented to make fast CU size determination in this section.

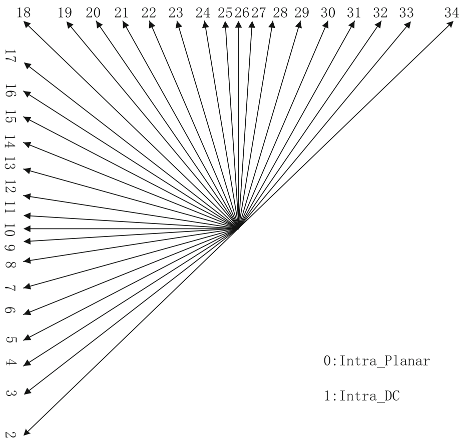


Fig. 2 Intra prediction modes in HEVC

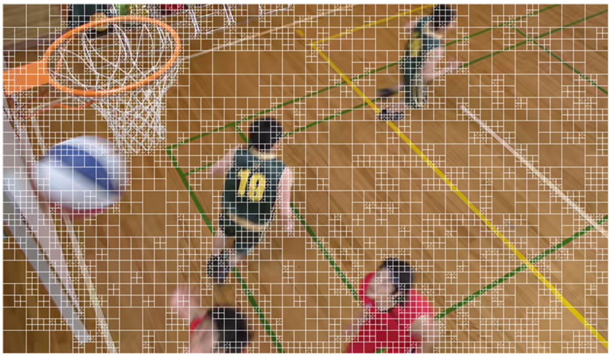


Fig. 3 CU splitting results of the intra predicition for the BasketballDrill sequence in the first frame

In our previous work in [14], the density and magnitude of the dissimilarities between the adjacent pixels are measured by the total variation (TV) of the H.264 MB. The TV of a MB is defined as (1).

$$TV_{MB} = \|\nabla I_{MB}\|_{TV} = \sum_{i=0}^{14}\sum_{j=0}^{14}\sqrt{(I[i,j]-I[i+1,j])^2 + (I[i,j]-I[i,j+1])^2}. \tag{1}$$

where *I[i, j]* is the luma intensity of the pixel at *i*, *j* of the MB. The calculation of the TV includes multiplication and square root operations. The DTV of a MB formulated in (2) can be employed as a replacement for simplifying calculating.

$$\begin{aligned} DTV_{MB} &= \|\nabla_i I_{MB}\|_{TV} + \left\|\nabla_j I_{MB}\right\|_{TV} \\ &= \sum_{i=0}^{14}\sum_{j=0}^{14}(|I[i,j]-I[i+1,j]| + |I[i,j]-I[i,j+1]|). \end{aligned} \tag{2}$$

Based on this concept, we utilize the DTV of the CU to determinate the complexity of textures for the treeblock of HEVC in this paper. The DTV of a CU DTV_{CU} formulated in (3)

$$\begin{aligned} \mathrm{DTV_{CU}} &= \|\nabla_i I_{CU}\|_{TV} + \left\|\nabla_j I_{CU}\right\|_{TV} \\ &= \sum_{i=0}^{N}\sum_{j=0}^{N}(|I[i,j]-I[i+1,j]| + |I[i,j]-I[i,j+1]|) \end{aligned} \tag{3}$$

Here, *I[i, j]* is the luma intensity of the pixel at *i*, *j* of the CU, *N* is the height or width of the CU.

The DTV of a CU measures the complexity of the CU. The larger the DTV_{CU} is, the denser and more significant the dissimilarities between pixels are and the more complex the current CU is. The smaller DTV_{CU} is, the more similarities between adjacent pixels in the CU and the fewer textures in the CU. According to the DTVs of CUs, we classify CUs into three types as follows:

$$Type_{cu} = \begin{cases} I\ , & DTV_{\mathrm{CU}} < = T1 \\ II\ , & T1 < DTV_{CU} <= T2 \\ III\ , & DTV_{CU} > T2 \end{cases} \tag{4}$$

where *T1* and *T2* are thresholds. The type of a CU decides the splitting strategy of this CU. The type *I* denotes this CU is not needed to further split, so the CU size determination is early terminated. The type *II* means that texture-complexity of the current CU is hard to determine, thus we execute the normal process without any further processing. The type *III* indicates the current CU is the highly rich texture region, then we split the CU into four sub-blocks directly, namely the current depth level is skipped.

Obviously, the thresholds *T1* and *T2* determinate the CU type classification. They are crucial for the CU size decision and have a strong impact on the coding efficiency. Here, we suggest an adaptive scheme to decide the thresholds. We choose the first frame in a group of picture (GOP) as the key frame to obtain the optimal thresholds. All the CUs in the key frame calculate the DTVs as (3) and are coded in normal way. Then, the DTVs are stored to the corresponding buffer according to the splitting strategy of the current sized CU. Three buffers are provided in the proposed method to store the DTVs of CUs that will not be split further in the prediction process, named DTV_buf_64_0, DTV_buf_32_0 and DTV_buf_16_0 corresponding to 64 × 64, 32 × 32 and 16 × 16 CUs, respectively. Analogously, other three buffers named DTV_buf_64_1, DTV_buf_32_1 and DTV_buf_16_1 are defined for DTVs of 64 × 64, 32 × 32 and 16 × 16 CUs. that will be split into smaller CUs in the prediction process. The DTVs of the 8 × 8 PUs type with the 2N×2N mode and N×N mode are stored in the DTV_buf_8_0 and DTV_buf_8_1, respectively. For example, supposing the current CU size is 32 × 32, if this CU is split in the prediction process, then the DTV of the current CU is saved in the DTV_buf_32_1. Otherwise, the DTV is stored in the DTV_buf_32_0. After the first frame is finished, the maximum DTV of the N×N CU for the DTV_buf_N_1 and the minimal DTV for the DTV_buf_N_0 are obtained. We denote them as TH_N_0, TH_N_1, respectively. Here, N represents the CU size which should be 64, 32, 16 or 8. Then we calculate the thresholds for CU classification according to TH_N_0 and TH_N_1. We set the thresholds T1 and T2 for each CU size empirically based on our tests on lots of sequences with various resolutions as follows:

$$\mathrm{TH}_{N\times N} = (\mathrm{TH}\_N\_0 + TH\_N\_1)/2 \tag{5}$$

$$T1_{\mathrm{N}\times N} = TH_{\mathrm{N}\times N} \tag{6}$$

$$T2_{\mathrm{N}\times N} = TH_{\mathrm{N}\times N}/4 \tag{7}$$

Here, N×N is the size of CU.

As described above, the key frame is coded in the normal way to obtain the thresholds for the CU type classification. The proposed adaptive DTV threshold-based CU size decision is employed in the successive frames of the current GOP. For each CU of these frames, we calculate the DTV_{CU} firstly and classify the CU types following (4). Then, we make the splitting decision for each CU according to its type. Note that, to maintain the adaptability of the thresholds, we refresh the DTV buffers and recalculate the type classification thresholds for each GOP. Namely, the thresholds setting process need to be executed in the first frame (key frame) of every GOP to make sure the thresholds can be updated as the video content.

3.2 Orientation gradient–based mode decision

As discussed in Section 2, after CU splitting, HEVC firstly selects N candidate modes in 35 prediction modes by the RMD process, N for size 64 × 64, 32 × 32, 16 × 16, 8 × 8, and 4 × 4 are

3, 3, 3, 8, and 8, respectively. Then, RDO is performed on the candidate mode set to obtain the best mode. Considering the spatial correlation of the prediction modes, the optimal mode of the current block is very likely to be the mode adopted in the neighbor block. Thus, the MPM are added into the candidate mode set for RDO. Therefore, we must traverse 35 prediction modes in the RMD process to choose the first N candidate modes and go through N+M modes for RDO. The computation complexity of the encoder is very large. We employ a orientation gradient-based model to reduce the number of directional prediction modes involved in the RMD and the candidate mode set size in the RDO simultaneously. The orientation gradient of a directional prediction mode is defined as (8):

$$\mathrm{G(m)}=\left(\sum_{i=1}^{n}\left(\frac{Ang_m}{32}\times|R_{i+1}-I_i|+\frac{(32-Ang_m)}{32}\times|R_i-I_i|\right)\right)/n \tag{8}$$

Where m denotes the number of the directional prediction mode. I_i is the original pixel value whose pixel is down-sampled from the current CU, n means the element number of down-sampled pixels. R_{i+1} and R_i are the reconstructed pixel values of the adjacent reference pixels of the current pixel I_i in the directional prediction mode m, $|R_{i+1}-I_i|$ and $|R_i-I_i|$ represent the absolute values of prediction residuals. Each predicted sample is obtained by projecting its location to a reference row of pixels in the selected prediction direction and then interpolating a value of the sample with 1/32 pixel accuracy. So, we divide the prediction residuals by 32 in the model. Ang_m represents the absolute value of the prediction angle of the mode m. Table 1 shows the prediction angles corresponding to the prediction modes.

In order to reduce computational load, we just sample a few representative pixels in a PU to calculate the orientation gradients. For a $N\times N$ PU, we sample four representative pixels on each 4×4 block, so $N\times N/4$ representative pixels are obtained. The four representative pixels for a 4×4 block corresponding to each directional prediction mode as showed in Fig. 4, which (a) relates to modes 2 to 9, (b) corresponding to 10 to 17, (c) for 18 to 25, and (d) for 26 to 34. The orientation gradient of each mode is calculated by (8). For example, Fig. 5 shows the choices of the reference pixels with four sampled pixels ($n=4$) corresponding to mode 2, 12, 22, 32. The orientation gradients of mode 2, 12, 22, 32 are calculated by (9), (10), (11), (12), respectively. Here, the bit shift is employed instead of division to accelerate the computation.

$$\mathrm{G}_{Mode=2}=\left(\left|R_L-I_f\right|+\left|R_M-I_j\right|+|R_M-I_d|+|R_N-I_k|\right)>>2 \tag{9}$$

$$\begin{aligned}\mathrm{G}_{\ Mode=12}=&\left(\left(\frac{5}{32}\left|R_{\mathrm{I}}-I_f\right|+\frac{27}{32}\left|R_{\mathrm{J}}-I_f\right|\right)+\left(\frac{5}{32}\left|R_{\mathrm{I}}-I_g\right|+\frac{27}{32}\left|R_{\mathrm{J}}-I_g\right|\right)\right.\\&\left.+\left(\frac{5}{32}\left|R_{\mathrm{J}}-I_j\right|+\frac{27}{32}\left|R_{\mathrm{K}}-I_j\right|\right)+\left(\frac{5}{32}\left|R_{\mathrm{K}}-I_p\right|+\frac{27}{32}\left|R_{\mathrm{L}}-I_p\right|\right)\right)>>2\end{aligned} \tag{10}$$

Table 1 Prediction angles corresponding to prediction modes

Mode	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Angle	32	26	21	17	13	9	5	2	0	−2	−5	−9	−13	−17	−21	−26	−32
Mode	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
Angle	−26	−21	−17	−13	−9	−5	−2	0	2	5	9	13	17	21	26	32	

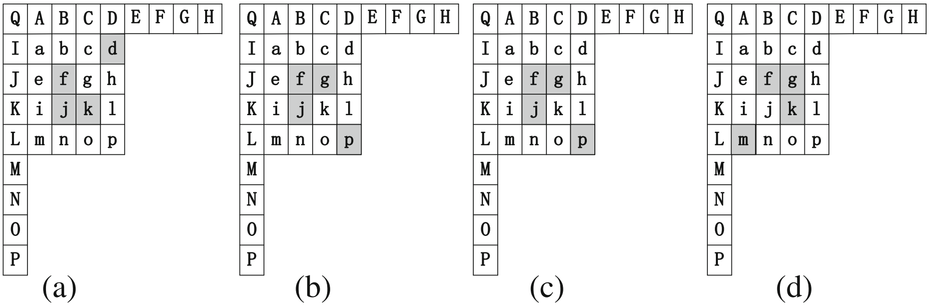


Fig. 4 The sample representative pixels for the 4 × 4 block for the directional prediction mode, **a**–**d** responding to mode from 2 to 9, from 10 to 17, from 18 to 25, from 26 to 34, respectively

$$\begin{aligned}\mathrm{G}_{Mode=22}=&\left(\left(\frac{13}{32}\left|R_{\mathrm{Q}}-I_j\right|+\frac{19}{32}\left|R_{\mathrm{I}}-I_j\right|\right)+\left(\frac{13}{32}\left|R_{\mathrm{A}}-I_f\right|+\frac{19}{32}\left|R_{\mathrm{B}}-I_f\right|\right)\right.\\&\left.+\left(\frac{13}{32}\left|R_{\mathrm{B}}-I_{\mathrm{g}}\right|+\frac{19}{32}\left|R_{\mathrm{C}}-I_g\right|\right)+\left(\frac{13}{32}\left|R_{\mathrm{B}}-I_p\right|+\frac{19}{32}\left|R_{\mathrm{C}}-I_p\right|\right)\right)>>2\end{aligned}\tag{11}$$

$$\begin{aligned}\mathrm{G}_{Mode=32}&\left(\left(\frac{21}{32}\left|R_D-I_f\right|+\frac{11}{32}\left|R_C-I_f\right|\right)+\left(\frac{21}{32}\left|R_E-I_g\right|+\frac{11}{32}\left|R_D-I_g\right|\right)\right.\\&\left.+\left(\frac{21}{32}\left|R_E-I_k\right|+\frac{11}{32}\left|R_D-I_k\right|\right)+\left(\frac{21}{32}\left|R_D-I_m\right|+\frac{11}{32}\left|R_C-I_m\right|\right)\right)>>2\end{aligned}\tag{12}$$

After calculating orientation gradients of 33 directional prediction modes, 10 modes with the minimum gradients are selected as the candidate modes. Adding the DC mode and PLANAR mode, a total of 12 modes are involved in the RMD process. Thus, we reduce the number of the candidate mode set from 3, 3, 3, 8, and 8 to 2, 2, 2, 5, and 5 for PU size 64 × 64, 32 × 32, 16 × 16, 8 × 8, 4 × 4, respectively. In final, RDO is performed on the candidate mode set including the MPM to search the optimal mode. The proposed orientation gradient-based mode decision algorithm is summarized as the algorithm 1.

Algorithm 1: Fast Mode Decision Algorithm

```
1. for $1<m<35$
```

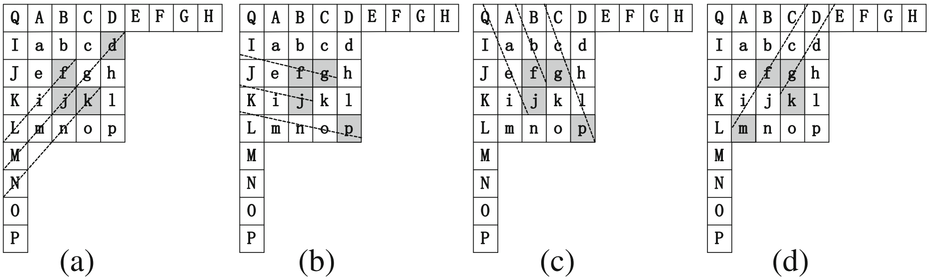


Fig. 5 Choices of reference pixels with four sampled pixels, **a**–**d** responding to mode 2, 12, 22, 32, respectively

```
  2. G (m) ← $\left(\sum_{i=1}^{n}\left(\frac{Ang_m}{32}\times\left|R_{i+1}-I_i\right| + {}^{(32-Ang_m)}\, 32\times\left|R_i-I_i\right|\right)\right)/n$
  3. end for
  4. select 10 kinds of mode with the minimum $G(m)$ into set S,
S = {$m_1$, $m_2$, $m_3$, …, $m_{10}$}
  5. S = $S$ + {$DC$, $PLANAR$}
  6. execute the RMD on $S$, $S_0$ is the candidate mode set by the RMD,
$S_0 \in S$, the number of $S_0$ is 2, 2, 2, 5, 5 for PU size 64 × 64, 32 × 32,
16 × 16, 8 × 8, 4 × 4, respectively
  7. $S_0 = S_0$ + MPM
  8. Execute the RDO on $S_0$ to choose the optimal mode
```

4 Experimental results

In this section, we evaluate the performance of the proposed method. The state-of-art methods proposed in [5, 13, 19] are chosen as benchmarks. All the fast mode decision algorithms are implemented on the newest version of the HEVC reference model HM 10.0 [4]. Twenty representative sequences from the official common HM test conditions and software reference configurations [1] are picked for testing. All the testing sequences are classified by their resolutions to five classes as Class A, Class B, Class C, Class D, and Class E, corresponding to the resolution of 2560 × 1600, 1920 × 1080, 832 × 480, 416 × 240, and 1280 × 720 respectively. For each testing sequence, 100 frames are coded and all the results are averaged over all the coded frames. The experiments are executed under all the intra setting of Main Profile with QPs 22, 27, 32, and 37. The performance gain or loss of the fast algorithms is measured in terms of the peak signal noise ratio (PSNR) loss of the Y component of the reconstructed image, the bit rate (BR) increase and the complexity reduction (measured by time saving) with respect to the HM full search scheme, denoted as ΔP, ΔBR and ΔTS, respectively. The measurements are all averaged across four QPs as follows.

$$\Delta P=\frac{1}{4}\sum_{i=1}^{4}\mathrm{Y}\_PSNR_{HM10.0}(QP_i)-Y\_PSRN_{\mathrm{proposed}}(QP_i) \tag{13}$$

$$\Delta\mathrm{BR}=\frac{1}{4}\sum_{i=1}^{4}\frac{BiteRate_{proposed}(QP_i)-BitRate_{HM10.0}(QP_i)}{BitRate_{HM10.0}(QP_i)}\times 100\% \tag{14}$$

$$\Delta\mathrm{TS}=\frac{1}{4}\sum_{i=1}^{4}\frac{Time_{HM10.0}(QP_i)-Time_{\mathrm{P}roposed}(QP_i)}{Time_{HM10.0}(QP_i)}\times 100\% \tag{15}$$

Where $\mathrm{QP_i}$ is the QP which can be 22, 27, 32, and 37. $\mathrm{Y\_PSNR_{HM10.0}}(QP_i)$, $\mathrm{BiteR}ate_{\mathrm{HM10.0}}(QP_i)$, $\mathrm{Time_{HM10.0}}(QP_i)$ denote the PSNR of the Y component, BR, coding time in HM 10.0 with $\mathrm{QP_i}$, respectively. $\mathrm{Y\_PSNR_{proposed}}(\mathrm{QP}_i)$, $\mathrm{BiteR}ate_{\mathrm{proposed}}(QP_i)$, $\mathrm{Time_{proposed}}(QP_i)$ depict the PSNR of the Y component, BR, coding time of the proposed method with $\mathrm{QP_i}$, respectively.

To evaluate the efficiency of the propose methods, we will assess the adaptive DTV threshold-based CU size determination method and the orientation gradient-based mode decision algorithm separately, denoted as Prop.-CU and Prop.-mode respectively. Meanwhile, the proposed size determination method and the mode decision method are combined together as a two stages algorithm, denoted as Prop.-Overall to assess the overall performance. Since coding is performed under all the intra prediction setting, the GOP size has no impact on the testing methods except the proposed CU determination method and the overall algorithm. Larger GOP sizes lead to more time saving but introduce larger coding performance loss too. Smaller GOP sizes can maintain the coding performance well but the time saving is limited. In this paper, the GOP size for CU size determination method is set to 30 empirically based on extensive experiments on various sequences and various QPs. This value can achieve a good tradeoff between the time saving and the coding performance loss. All the experimental results of different mode search strategies for testing sequences with various resolutions are detailed in Table 2. And the winners for each sequence in terms of ΔP, ΔBR and ΔTS are identified as bold in the table. The average efficiency gains or losses across all testing sequences are listed at the last line of the table.

As demonstrated in Table 2, Jiang's mode decision method [5] achieves a close coding performance to HM 10.0. For some sequences, it even outperforms all other methods in terms of the reconstructed image's PSNR. On average, the PSNR degradation and BR increase of this method are 0.044 dB and 0.79 %, respectively. But, the coding time is only saved 24.06 % which is the least of all the methods. The mode decision method presented in [19] saves about 29 % coding time, while the coding performance loss is some high. The average PSNR degradation and BR increase of this method across all testing sequences with respect to HM 10.0 are 0.074 dB and 1.05 %, respectively. The proposed mode decision method Prop.-Mode has a same average PSNR loss with Yang's method while its BR increase is significant less. This indicates the prediction mode filtering model proposed in this paper is more accurate than those proposed in [5] and [19]. And since there are fewer prediction modes selected for RMD and RDO, the proposed orientation gradient-based mode decision method is more efficient where the time saving gain put the 5 % in parentheses. Moreover, as listed in the Table 2, the proposed CU fast determination method Prop.-CU achieves the best reconstructed image quality among all the testing methods, in which only 0.035 dB PSNR is lost with respect to HM 10.0 with a BR increase of 0.83 % on average. And 42.20 % of the coding time is saved. This demonstrates the effectiveness and efficiency of the proposed adaptive DTV threshold-based CU determination model.

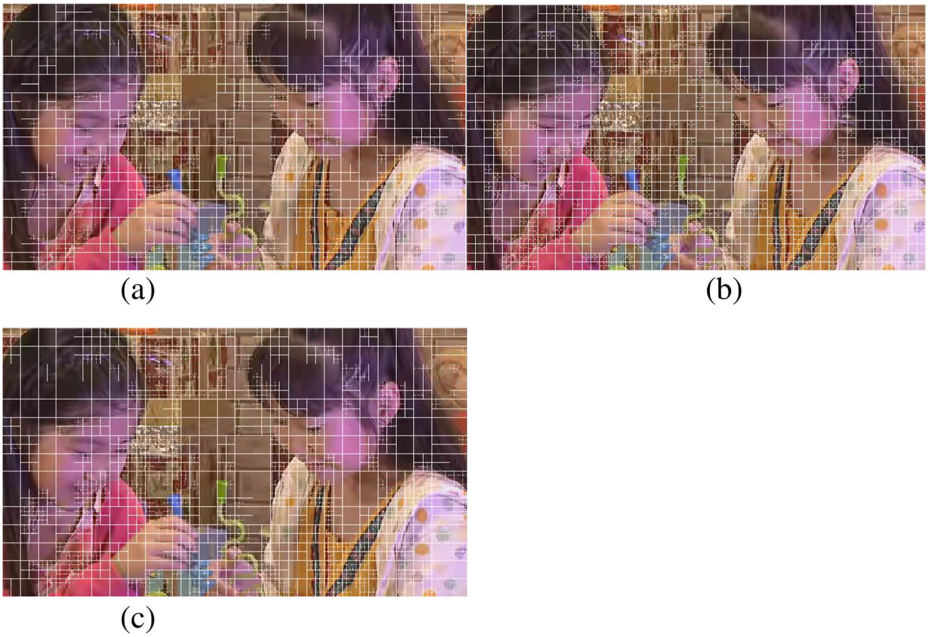
Compared with methods only applying the fast CU size determination or the fast mode decision, two-stage algorithms integrating the CU splitting early termination and mode filter generally save more coding time with some more RD performance loss. Shen et al. [13] proposed an overall algorithm where the fast CU size determination and fast mode decision are both employed. The Table 2 indicates that the overall proposed in [13] reduces the coding time 36.72 % with 0.122 dB PSNR degradation and 0.945 % BR increase with respect to HM 10.0. While based on the effective and efficient CU splitting early termination and mode filter models, the proposed overall algorithm Prop.-Overall saves 57.21 % coding time with only 0.079 dB and 0.88 % BR increase with respect to HM 10.0. The main performance gain stems from the proposed CU splitting termination strategy. Figure 6 shows the CU splitting results of the second frame of the BlowingBubbles sequence in Class D using HM 10.0, Shen's algorithm [13] and the proposed DTV threshold-based method. We choose the second

Table 2 Performance comparison between the proposed methods and the existing algorithms for sequence with various resolutions

Class & Sequences		[5]			[19]			[13]			Prop.-CU			Prop.-Mode			Prop.-Overall		
		ΔP	ΔBR	ΔTS	ΔP	ΔBR	ΔTS	ΔP	ΔBR	ΔTS	ΔP	ΔBR	ΔTS	ΔP	ΔBR	ΔTS	ΔP	ΔBR	ΔTS
		[dB]	[%]	[%]	[dB]	[%]	[%]	[dB]	[%]	[%]	[dB]	[%]	[%]	[dB]	[%]	[%]	[dB]	[%]	[%]
A	Traffic	0.036	0.48	25.76	0.066	0.71	29.18	0.105	0.49	39.01	**0.031**	1.09	37.42	0.041	**0.13**	31.24	0.073	1.00	**52.46**
	PeopleOnStreet	0.043	0.47	25.19	0.071	1.06	27.72	0.1	0.71	34.79	**0.037**	0.95	40.98	0.045	**0.08**	31.32	0.084	0.95	**55.34**
	NebutaFestival	**−0.045**	2.15	17.31	0.014	0.97	30.08	0.289	3.69	41.15	0.021	0.26	45.12	−0.012	**0.10**	26.77	0.012	0.37	**63.91**
	SteamLocomotiveTrain	**−0.052**	1.29	18.35	0.013	1.19	29.79	0.269	5.26	40.87	0.048	1.30	45.79	−0.014	**0.26**	26.53	0.045	1.83	**62.76**
B	ParkScene	0.042	0.43	26.90	0.057	1	30.42	0.101	**0.01**	40.63	**0.013**	0.25	34.73	0.040	0.45	31.83	0.051	0.15	**49.69**
	Kimono1	**0.007**	0.43	27.29	0.020	0.91	29.70	0.047	0.49	65.58	0.019	1.71	22.68	0.009	**0.05**	31.72	0.028	1.60	**42.58**
	Cactus	0.041	0.61	25.92	0.058	0.69	30.14	0.083	0.23	39.49	**0.020**	**0.09**	37.04	0.037	0.17	31.50	0.057	0.10	**51.28**
	BQTerrace	0.043	0.31	24.99	0.070	0.44	31.38	0.089	0.53	42.25	**0.030**	0.51	40.57	0.047	**0.11**	32.23	0.074	0.35	**55.84**
	BasketballDrive	0.017	1.45	26.46	0.029	0.94	30.12	0.04	1.13	43.81	0.026	0.60	58.06	**0.016**	**0.16**	31.71	0.035	0.71	**64.63**
C	BQMall	0.071	0.37	22.67	0.103	0.99	28.58	0.127	0.36	31.6	**0.060**	1.03	38.74	0.060	**0.09**	29.83	0.122	0.94	**55.93**
	BasketballDrive	0.051	0.34	23.17	0.079	1.41	28.25	0.083	0.19	29.72	0.063	0.37	50.13	**0.043**	**0.06**	29.63	0.100	0.48	**62.25**
	ParkScene	0.118	0.53	23.60	0.158	0.99	29.17	0.179	0.01	30.83	**0.031**	**0.10**	37.10	0.108	0.17	29.06	0.126	0.02	**53.99**
	RaceHorses	0.061	0.28	23.73	0.082	0.72	28.70	0.109	**0.03**	26.68	**0.010**	1.32	40.41	0.057	0.22	29.47	0.080	0.97	**56.13**
D	BasketballPass	0.059	1.3	24.07	0.085	1.15	28.94	0.1	0.31	22.76	**0.043**	0.38	46.88	0.045	**0.16**	30.59	0.082	0.58	**58.84**
	BlowingBubbles	0.082	0.51	23.41	0.113	0.94	29.43	0.14	0.02	25.45	**0.008**	0.66	32.81	0.076	**0.02**	30.18	0.092	0.50	**52.75**
	BQSquare	0.115	0.84	24.35	0.165	2.03	30.71	0.179	0.4	26.66	**0.052**	**0.18**	46.16	0.114	0.24	29.09	0.144	0.62	**61.38**
	RaceHorses	0.084	0.8	24.15	0.110	1.17	28.88	0.134	0.22	25.25	**0.013**	1.10	35.56	0.072	**0.09**	30.35	0.095	1.09	**55.02**
E	FourPeople	**0.038**	0.56	24.34	0.072	1.13	28.53	0.099	0.91	35.66	0.062	1.02	48.97	0.039	**0.09**	30.46	0.107	1.06	**61.71**
	Johnny	0.026	1.55	25.14	0.044	1.44	29.42	0.076	2.01	47.8	0.038	1.79	47.97	**0.022**	**0.28**	30.88	0.064	2.08	**60.60**
	KristenAndSara	0.041	1.06	24.37	0.066	1.16	28.88	0.101	1.9	44.39	0.069	1.82	56.83	**0.037**	**0.15**	30.77	0.103	2.17	**67.03**
Average		0.044	0.79	24.06	0.074	1.05	29.40	0.1225	0.945	36.72	**0.035**	0.83	42.20	0.044	**0.15**	30.26	0.079	0.88	**57.21**

frame because of the first frame in the proposed method is the key frame. As seen in Fig. 6, all three methods tend to larger block sizes in homogeneous areas while smaller block sizes are used in regions with rich textures more often. However, our CU splitting results are more consistent with HM 10.0 than [13]. Besides, most of differences of the optimal depths between the proposed algorithm and HM 10.0 are only 1, and no more than 2. So, the proposed method achieves a better RD performance than Shen's method.

For a more intuitive representation of the experimental results, Fig. 7 presents the RD-curves and the encoding time comparison for the BQTerrace sequence in Class B of Jiang's method [5], Yan' method [19], Shen' method [13], Prop.-CU, Prop.-mode, Prop-overall and HM 10.0. Figure 7a illustrates the detailed RD performance for all testing methods under four QPs 22, 27, 32, and 37. It can be observed from Fig. 7a that Prop.-CU and Prop.-Mode are the closest curves to that of HM 10.0. That is, smaller RD performance loses are introduced by these two proposed methods than other competitive algorithms. And as indicated in Fig. 7a, the proposed overall algorithm achieves a competing RD performance too. The RD curve of it is above those of Shen's and Yang's method. Figure 7b details the average coding time for per frame of all the methods for the BQTerrace sequence under various QPs. As shown in Fig. 7b, the proposed overall algorithm outperforms all the other methods a lot for all four QPs. And Prop.-CU and Prop.-Mode also attain more time saving gains than methods proposed by Jiang [5] and Yan [19]. Prop.-CU even needs almost less coding time than the overall method proposed by Shen [13] under smaller QPs for this sequence.


(a)

(b)

(c)

Fig. 6 CU splitting results on different fast CU size decisions with the BlowingBubbles sequence in the second frame in Class D: **a** for HM 10.0, **b** for Shen's algorithm [13], **c** for proposed CU size decision

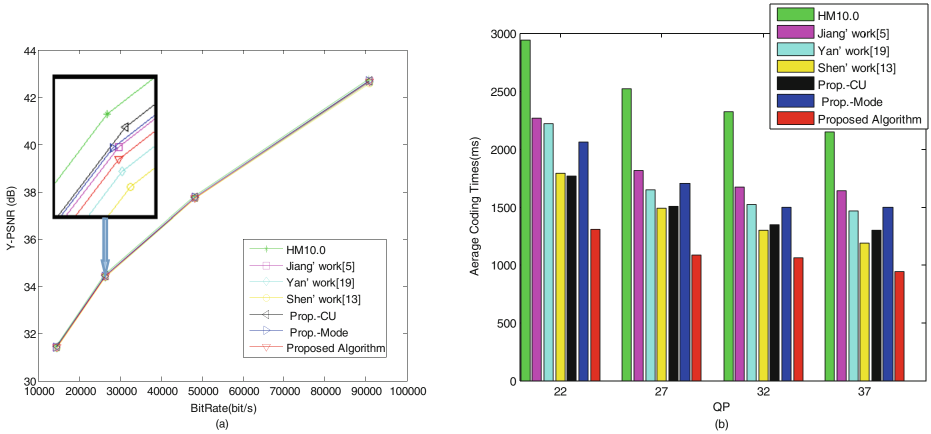


Fig. 7 Performances comparison on different fast algorithms with the BQTerrace sequence in Class B: **a** R-D curves, and **b** Encoding time

5 Conclusion

In this paper, we propose an adaptive DTV threshold-based CU size determination method and an orientation gradient-based mode decision method and combine them as a two-stage algorithm. In the CU splitting stage, we skip or early terminate some specific depth levels according to the CU type classified by the DTV of the CU. Then, some unlikely prediction modes are filtered before the RMD and RDO process based on the orientation gradients of the prediction modes. The experimental results demonstrate that the proposed methods outperform state-of-the-art fast CU size determination and mode decision algorithms. The two-stage algorithm saves coding time up to 57.21 % on average with a negligible PSNR loss (about 0.079 dB) and BR increase (about 0.88 %) with respect to HM 10.0. In future research, we will investigate the possibility of introducing the proposed CU side decision strategy into multi-view video coding [10].

Acknowledgments This work was supported in part by the Hunan Province Science and Technology Planning Project (nos. 2014FJ6047 and 2014GK3030), the Science Research Key Project of the Education Department of Hunan Province (nos. 13A107 and 15A007), and the Changsha Science and Technology Planning Project (no. K1403028-11).

References

1. Bossen F (2013) Common HM test conditions and software reference configurations. JCT-VC document L1100
2. Cho S, Kim M (2013) Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. IEEE Trans Circuits Syst Video Technol 23(9):1555–1564
3. Fini MR, Zargari F (2015) Two stage fast mode decision algorithm for intra prediction in HEVC. Multimed Tools Appl, p 1–18
4. HEVC Reference Model [Online]. Available: http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/
5. Jiang W, Ma H, Chen Y (2012) Gradient based fast mode decision algorithm for intra prediction in HEVC. Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on. IEEE, p 1836–1840

6. Kim J, Choe Y et al (2013) Fast coding unit size decision algorithm for intra coding in HEVC. Consumer Electronics (ICCE), 2013 I.E. International Conference on. IEEE, p 637–638
7. Kim I-K, Lee T et al (2012) Block partitioning structure in the HEVC standard. IEEE Trans Circuits Syst Video Technol 22(12):1697–1706
8. Lainema J, Bossen F, Han W-J et al (2012) Intra coding of the HEVC standard. IEEE Trans Circuits Syst Video Technol 22(12):1792–1801
9. Li Y, Liu Y, Yang H et al (2014) Fast CU splitting and pruning method based on online learning for intra coding in HEVC. Visual Communications and Image Processing Conference, 2014 IEEE. IEEE
10. Pan Z, Zhang Y, Kwong S (2015). Ecient motion and disparity estimation optimization for low complexity multiview video coding. IEEE Trans Broadcasting 61(2):166–176
11. Piao Y, Min JH, Chen J (2010) Encoder improvement of unified intra prediction. JCTVC-C207. JCT-VC of ISO/IEC and ITU-T, Guangzhou, China
12. Shen L, Liu Z, Zhang X et al (2013) An effective CU size decision method for HEVC encoders. IEEE Trans Multimedia 15(2):465–470
13. Shen L, Zhang Z, An P (2013) Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Trans Consum Electron 59(1):207–213
14. Song Y, Long J, Yang K et al (2014) Complexity scalable intra-prediction mode decision algorithm for mobile video applications. IET Commun 8(9):1654–1662
15. Song Y, Shen YF, Long JZ et al (2013) Intra-prediction mode decision algorithm based on orientation gradient for H.264. Chin J Comput 8:1757–1764
16. Sullivan GJ, Han W-J et al (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668
17. Vanne J, Viitanen M, Hamalainen TD et al (2012) Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs. IEEE Trans Circuits Syst Video Technol 22(12):1885–1898
18. Wiegand T (2003) Draft ITU-T recommendation and final draft international standard of joint video specification. ITU-T rec. H. 264| ISO/IEC 14496-10 AVC
19. Yan S, Hong L, He W, Wang Q (2012) Group-based fast mode decision algorithm for intra prediction in HEVC. Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on. IEEE, p 225–229
20. Zhang H, Ma Z (2014) Fast intra mode decision for high efficiency video coding (HEVC). IEEE Trans Circuits Syst Video Technol 24(4):660–668
21. Zhang M, Zhao C, Xu J (2012) An adaptive fast intra mode decision in HEVC. Image Processing (ICIP), 2012 19th IEEE International Conference on. IEEE, p 221–224
22. Zhao L, Zhang L, Ma S, Zhao D (2011) Fast mode decision algorithm for intra prediction in HEVC. Visual Communications and Image Processing (VCIP), 2011 IEEE. IEEE, p 1–4

Yun Song is an associate professor in the School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha, China. He is pursuing his PhD from Hunan University, China. His research interests include video compression, video processing and compressive sensing for image and video processing.

Ye Zhen is pursuing her master degree from the School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha, China. Her research interests include video compression and compressive sensing for video processing.

Xueyu Li is pursuing her master degree from the School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha, China. Her research interest includes video compression and compressive sensing for video processing.

Biye Cai an associate professor in the School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha, China. He is pursuing his PhD from Hunan University, China. His research interests include image processing and pattern recognition.

Gaobo Yang is a professor in Hunan University, China. He is also a key member of Hunan Provincial Key Laboratory of Networks and Information Security. He received the Ph.D. degree in Communication and Information System from Shanghai University in 2004. He is the PI of several projects such as Natural Science Foundation of China (NSFC), Special Pro-phase Project on National Basic Research Program of China (973) and program for New Century Excellent Talents (NCET) in university. Currently, his research interests are in the area of image and video signal processing, digital media forensics.