Signal Processing: Image Communication 47 (2016) 271–279

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

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

A fast inter coding algorithm for HEVC based on texture and motion quad-tree models

Fen Chen, Peng Li, Zongju Peng*, Gangyi Jiang, Mei Yu, Feng Shao

Faculty of Information Science and Engineering, Ningbo University, Ningbo, Zhejiang, China

ARTICLE INFO

Article history:
Received 30 January 2016
Received in revised form
25 June 2016
Accepted 6 July 2016
Available online 7 July 2016

Keywords:
High efficiency video coding (HEVC)
Fast inter coding
Motion property
Quad-tree models

ABSTRACT

In order to reduce the encoding complexity of the emerging high efficiency video coding (HEVC) standard, a fast inter coding algorithm for HEVC based on texture and motion quad-tree models is proposed. First, the coding units are classified into motion or static blocks according to a motion/static detection algorithm. Then, different encoding schemes are adopted for each type of block. In particular, we can make full use of statistical properties and temporal correlations to determine the depth range and prediction mode of static largest coding tree units (LCTUs). Moreover, fast coding unit size and mode decision are made by means of quad-tree models and spatiotemporal correlations for the motion LCTUs. Finally, we merge the above schemes together to gain better performance. Experimental results show that the proposed overall algorithm, compared with the original HEVC encoding scheme, reduces encoding time by 47.5% on average with a Bjonteggard delta bit rate increase of only 1.6% for various test sequences under random access condition. Compared with a state-of-the-art algorithm, the proposed method can save more encoding time while maintaining comparative rate distortion performance.

1. Introduction

The demand for high definition (HD) and ultra-HD (UHD) multimedia services in TV broadcasting, internet video, or even mobile applications is already imminent. However, the problems of high-bandwidth video transmission and large-capacity storage required by these HD and UHD videos have remained major research subjects in recent years. To address these problems, the Joint Collaborative Team on Video Coding (JCT-VC) developed the high efficiency video coding (HEVC) standard [1] in January 2013, which is a successor to H.264/AVC (Advanced Video Coding) [2]. The goal of HEVC is to achieve about a 50% bit-rate reduction over high profile H.264/AVC given the same objective video qualities [3].

The hierarchical quad-tree structure based on the coding tree unit (CTU) has been adopted in HEVC. The CTU is one of the most powerful tools available for improving HEVC coding efficiency. It includes the coding unit (CU), prediction unit (PU), and transform unit (TU) to describe the overall encoding process. HEVC intra prediction adopts a quad-tree structure based coding technique and multiple-angle intra prediction to improve coding efficiency, while the high-precision motion compensation based on variable PU size, adaptive motion vector prediction, and merging techniques are employed in the inter coding. These new techniques significantly improve the compression efficiency of HEVC [4]. However, they lead to intensive computational complexity, which hinders real-time applications of HD and UHD videos [5]. Therefore, it is important to reduce the complexity of the HEVC encoders as much as possible without video quality degradation.

HEVC inter coding makes full use of temporal correlation to reduce coding redundancy, further improving compression efficiency on the basis of intra coding. Because of the introduction of new techniques in HEVC, many fast algorithms designed for H.264/AVC cannot be applied to HEVC in a straightforward manner. Recently, there have been quite a few related methods aiming to reduce the complexity of the HEVC encoders. They can be roughly divided into three categories: fast CU size decision, fast mode decision, and combination of both. In this paper, we focus on combining fast decision for both CU size and prediction mode. Gweon et al. [6] proposed a fast coded block flag (CBF) algorithm that utilized the zero CBF of the current PU to early terminate the mode decision of the next PU. Cho et al. [7] proposed an early termination method based on the property of the Skip mode. Yang et al. [8] proposed an early Skip determination method based on motion vector difference and CBF. The above algorithms [6–8] take advantage of special encoding parameters to terminate the CU or PU mode decisions early, reducing the encoding complexity to some extent. However, the number of CUs satisfying the conditions of these special encoding parameters is small, the encoding time saving is quite limited.

* Corresponding author.
E-mail address: pengzongju@126.com (Z. Peng).

http://dx.doi.org/10.1016/j.image.2016.07.002
0923-5965/

There have been many investigations about fast mode decisions for inter prediction in HEVC. Shen et al. [9] presented an adaptive inter-mode decision algorithm for HEVC that jointly utilizes inter-level and spatiotemporal correlations. Based on statistical analyses, three methods were proposed: early Skip mode decision, prediction size correlation-based mode decision, and rate-distortion (RD) cost correlation-based mode decision. Lee et al. [10] put forward an early Skip mode decision method by utilizing its distortion characteristics after calculating the RD cost of a $2N \times 2N$ Merge mode. Ahn et al. [11] proposed a fast encoding scheme for HEVC inter coding which fully utilizes spatiotemporal encoding parameters to early determine CU size and Skip mode. This scheme uses sample-adaptive-offset parameters as the spatial encoding parameter to estimate the texture complexity and uses motion vectors, TU size, and CBFs as the temporal encoding parameters to measure motion complexity.

In fact, similarly to H.264/AVC, the most time-consuming computation in HEVC is the inter prediction process, which includes motion estimation (ME). Therefore, many fast ME algorithms have also been proposed to reduce inter-slice coding complexity. Jou et al. [12] proposed a fast ME algorithm aimed at real-time video encoding, which adopts a predictive integer ME algorithm that selects the most probable search direction and employs a PU size-dependent fractional ME to reduce interpolation filtering. Lee et al. [13] proposed a fast ME algorithm using a priority-based inter mode decision method. This algorithm decided whether to perform ME by calculating the priority of each mode, reducing the encoding complexity of ME process by up to 55.51%. In addition, there are many algorithms utilizing parallel computing to reduce the computational complexity of encoders. Yan et al. [14] proposed a parallel computing framework for HEVC CU partitioning tree decision on many-core processors. The framework achieves 11 and 16 times speedup for 1920×1080 and 2560×1600 video sequences respectively. Yan et al. [15] also proposed a parallel framework to decouple ME for different partitions on many-core processors, and 30 and 40 times speedup for 1920×1080 and 2560×1600 video sequences have been achieved. The above algorithms [6–15] mainly focus on texture characteristics, motion properties, spatiotemporal correlations, or statistical information about the CU size and prediction modes to speed up HEVC encoders. However, they do not make full use of these information, and most of them cannot consider the motion and edge information of different CU depth levels. Hence, there is still considerable space for improvement in HEVC encoders.

To overcome the limitations of the above algorithms, this paper proposes a fast inter coding algorithm based on texture and motion quad-tree models, considering the motion and edge information of each CU depth level. First, the proposed algorithm classifies the current Largest Coding Tree Unit (LCTU) into motion or static blocks according to a motion/static detection algorithm. Then, different fast encoding schemes are adopted for each type of block. A fast encoding scheme for static LCTUs is performed to determine their depth range and candidate mode set using statistical properties and temporal correlations of CU size and prediction modes. For the motion LCTUs, a fast CU splitting scheme is proposed that utilizes a texture quad-tree model (TQM) and temporal correlations of CU size, while the proposed fast mode decision scheme uses a motion quad-tree model (MQM) and spatial correlations of the best prediction mode. This paper has two main contributions: 1) We propose two quad-tree models (MQM and TQM) by fully utilizing the motion characteristics as well as the texture edge information of each CU depth level. Further, we propose an algorithm to early determine the probable depth range and mode set based on these two models so that we can speed up the encoding process with negligible loss in RD performance. 2) We develop different strategies for motion and static blocks based on the statistical experiments and analysis.

The rest of this paper is organized as follows. In Section 2, the HEVC inter coding complexity and statistics of static and motion LCTUs are analyzed in detail. In Section 3, we first briefly introduce the overall framework of the proposed algorithm, and then present two encoding schemes targeting static LCTUs and motion LCTUs in detail, respectively. The experimental results and analyses are given in Section 4. Finally, conclusions are made in Section 5 along with a summary of the proposed algorithm.

2. Inter coding complexity and statistical analyses in HEVC

In the HEVC standard, intra or inter prediction coding is performed on a quad-tree based block structure. The recursive block in HEVC is called the CTU, which can be 64×64, 32×32 or 16×16 in size. Of course, CTU can further split into four equally sized CUs, which is the leaf node of a quad-tree partitioning of the CTU. In the encoding process, the best combinations of CUs, PUs, and TUs are determined in the sense of RD optimization (RDO). In HEVC intra coding, each 8×8 PU can be divided into four partitions of size 4×4. Therefore, the original HEVC encoding scheme needs to calculate the RD-cost value 341 $(4^0+4^1+4^2+4^3+4^4=341)$ times in order to determine the best combination of CUs and PUs. As shown in Fig. 1, the LCTU recursively splits into sub-CUs from depth 0 to depth 3. Each CU can be further divided into PUs. The HEVC encoder has 12 different PU modes, including Skip mode, Merge mode, Inter_$2N \times 2N$, Inter_$2N \times N$, Inter_$N \times 2N$, Inter_$N \times N$, asymmetric modes (Inter_$N \times n$U, Inter_$2N \times n$D, Inter_nL $\times$ 2N, and Inter_nR $\times$ 2N), and Intra modes (Intra_$2N \times 2N$ and Intra_$N \times N$). Every PU except for the Intra PUs needs to perform ME to determine the best PU mode. The computational complexity of the ME process is quite high. In conclusion, CU size decision, inter mode decision, and the ME process lead to huge computational complexity of the Inter encoding process, limiting its use in real-time applications.

In order to design a fast HEVC encoding algorithm, we perform explorative experiments on the latest HEVC reference software (HM-16.9) and statistically analyze the experimental results. The Random Access-Main (RA-Main) profile was used and four QPs {22, 27, 32, and 37} are tested. RDO is enabled. The test sequences [16] provided by JCT-VC, including Traffic (2560×1600), Kimono (1920×1080), PartyScene (832×480), BlowingBubbles (416×240) and Johnny (1280×720), are used in the statistical experiment. We divide the test sequences set in two parts, and use one for the statistics and the other for the encoding. We find that the following phenomena and rules exist in HEVC.

(1) CU splitting results are different in various frames. Under the RA-Main test condition, a hierarchical B structure is used for encoding. The frame types include I frame, Generalized P and B (GPB) frames, reference B frames and non-reference B frames. Fig. 2 shows the optimal CU splitting of different frame types in the “BlowingBubbles” sequence. The CUs in the I and GPB frames have the smallest size, while those in the reference B slice have a relatively larger size, and those in the non-reference B slice have the largest size. The underlying reason of the phenomena is that the CU splitting is relative to the importance of these frames. I and GPB frames have the highest importance, followed by the reference B frames. Non-reference B frames have the least importance.
(2) Optimal depth distribution of static LCTUs in reference B frames is different from that in non-reference under different QPs. The extraction method of static LCTUs will be introduced in the succeeding section. The statistical results is shown in Fig. 3. The optimal depth range of most static LCTUs in both

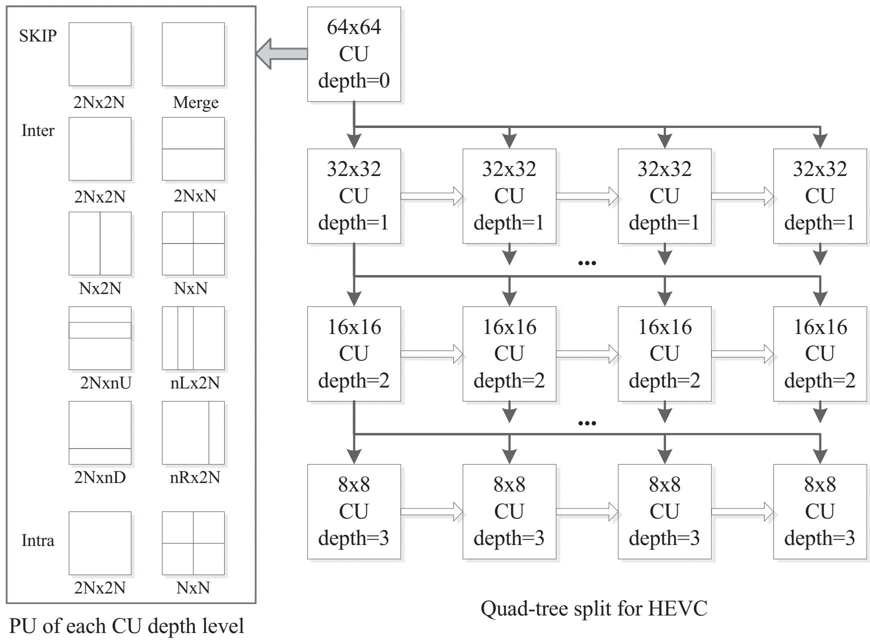
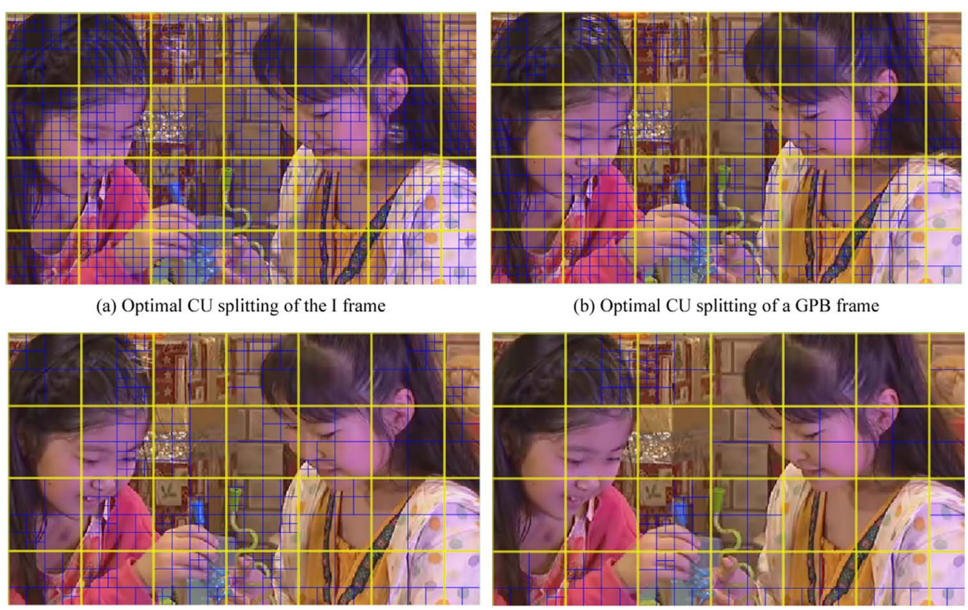


Fig. 1. Schematic diagram of quad-tree split and prediction modes for HEVC.



(a) Optimal CU splitting of the I frame
(b) Optimal CU splitting of a GPB frame
(c) Optimal CU splitting of a reference B frame
(d) Optimal CU splitting of a non-reference B frame

Fig. 2. Optimal CU splitting results of various frames in the “BlowingBubbles” sequence.

reference B frames and non-reference B frames is 0–2. In HEVC inter coding, the motion or static characteristic has a crucial effect on CU splitting and mode decision. Slight motion or static CUs usually select large block sizes as the optimal coding mode, while fast motion CUs usually select small block sizes. The CUs of different frame types and QP values have different depth distributions. In lowest CU depth level, the distribution proportion of non-reference B frames is higher than that reference B frames. It is consistent with the distribution in Fig. 2. When QP rises, more CUs are encoded with lower depth level. These features of CU depth distribution are utilized for design the proposed fast algorithm. Hence, the HEVC encoder can skip the ME operation for small CU sizes for these CUs.

(3) The optimal depth distribution of static or non-edge LCTUs is imbalance. The non-edge LCTUs is extracted by Sobel operator which will be detailed in the next section. Fig. 4 shows the statistical results of optimal depth levels of CU splitting. The percentage of static or non-edge LCTUs with depths ranging from 0 to 1 is 83.9% on average, thus we have no need to encode depth values 2 and 3.

(4) The CU splitting and mode selection of static LCTUs have strong temporal correlation because of the strong temporal correlations in video signal. Figs. 5 and 6 respectively show optimal mode distribution and depth distribution of static LCTUs when the optimal mode of the collocated LCTU is Skip. According to these figures, when the optimal mode of the collocated LCTU is Skip, the percentage of current static LCTUs for which the best mode is Skip is 90.4% on average and those for which the optimal depth of the current LCTU is zero is 90.4% on average. Therefore, the CU size and mode

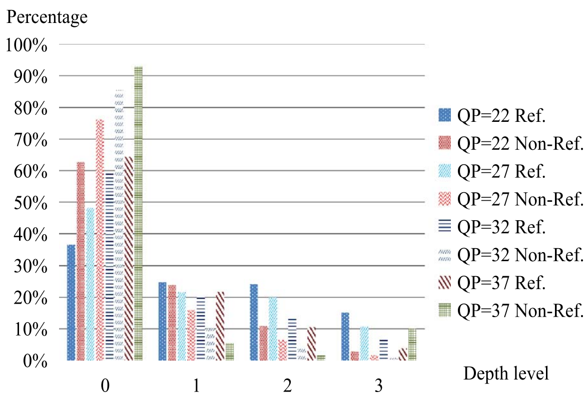


Fig. 3. Optimal depth distribution of static LCTUs among reference and non-reference frames.

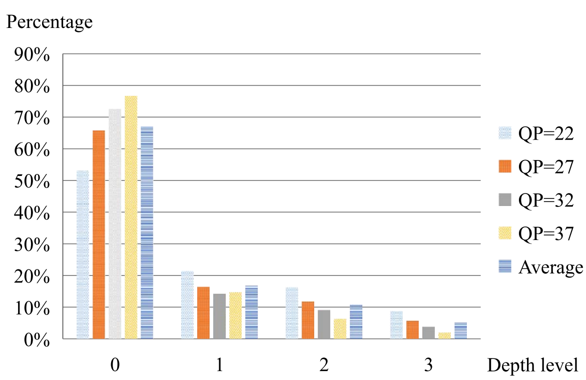


Fig. 4. Optimal depth level distribution of static or non-edge LCTUs.

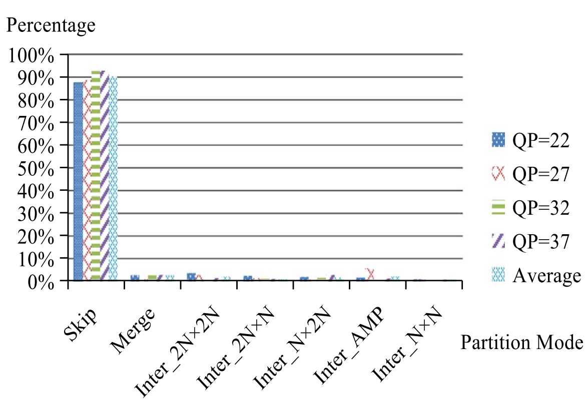


Fig. 5. Optimal mode distribution of static LCTUs when the optimal mode of the collocated LCTU is Skip.

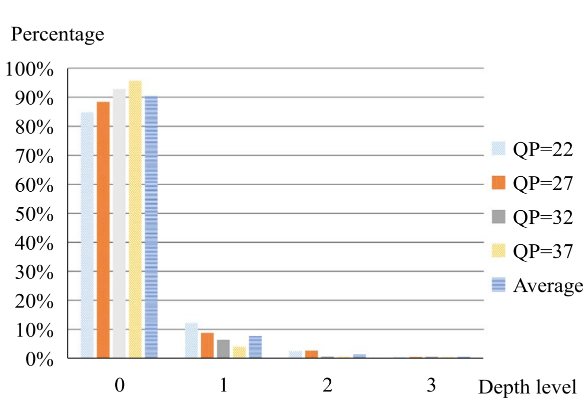


Fig. 6. Optimal depth distribution of static LCTUs when the optimal mode of the collocated LCTU is Skip.

distribution of static LCTUs can refer to the optimal CU size and mode of collocated CUs in temporal reference frames.

(5) The CU splitting and mode selection of motion LCTUs are different from those of static LCTUs. For low-resolution video sequences with complex texture, the percentage of motion CUs is extremely high. The motion LCTUs appear the following features. First, the optimal size of motion CUs is closely related to texture edge characteristics. Second, the optimal size of a motion CU is usually small. Third, motion LCTUs contain many static sub-CUs that are likely to select Skip or Merge mode as the best mode. Hence, we build TQM and MQM and design fast CU splitting and mode decision strategies for motion LCTUs according to these features.

According to the above observations and statistical analyses, this paper presents a fast inter coding algorithm for HEVC based on texture and motion quad-tree models. In the proposed algorithm, we only optimize the encoding process of B frames, and I and GPB frames are encoded using original HM scheme. The reasons for this strategy are as follows. On one hand, the encoding distortion of I and GPB frames will propagate to the reference frames. Intra prediction has a greater impact on subjective video quality than inter prediction [17]. On the other hand, inter prediction consumes about 60%–96% of the encoding time of the overall encoding process.

The proposed algorithm first classifies current LCTU of the B frames into motion or static LCTUs according to the motion/static detection algorithm. Then, two fast encoding schemes, denoted by Scheme I and Scheme II, are respectively adopted for static and motion LCTUs. The fast encoding scheme for static LCTUs contains depth range determination and early Skip mode determination. The fast encoding scheme for motion LCTUs contains fast CU splitting, preliminary mode decision, and early Skip/Merge determination.

3. Proposed fast inter coding algorithm for HEVC

In this section, we firstly briefly introduce the overall framework of the proposed algorithm, and then different encoding strategies are detailed consecutively.

3.1. Overall flowchart of the proposed algorithm

Fig. 7 shows the overall flowchart of the proposed fast inter coding algorithm in this paper, which is summarized as follows:

Step 1: All LCTUs in the B frames are categorized into motion LCTUs and static LCTUs by the motion/static LCTUs detection algorithm. For each LCTU, the following Steps are conducted.
Step 2: If the current LCTU is a static LCTU, go to Step 3. Otherwise, go to Step 4.
Step 3: For a static LCTU, the greatest possible depth range of the current LCTU is determined by considering the texture edges, frame type, and QP value on CU splitting. Early Skip mode determination of the static LCTU is then performed by utilizing temporal correlations.
Step 4: For a motion LCTU, the optimal CU size is determined by the TQM and temporal correlation of CU size. Early Skip/Merge mode determination and preliminary mode decisions for the motion CUs are performed based on the MQM and best modes of the spatially adjacent CUs.
Step 5: The optimal CU size and PU mode for the current LCTU are determined.

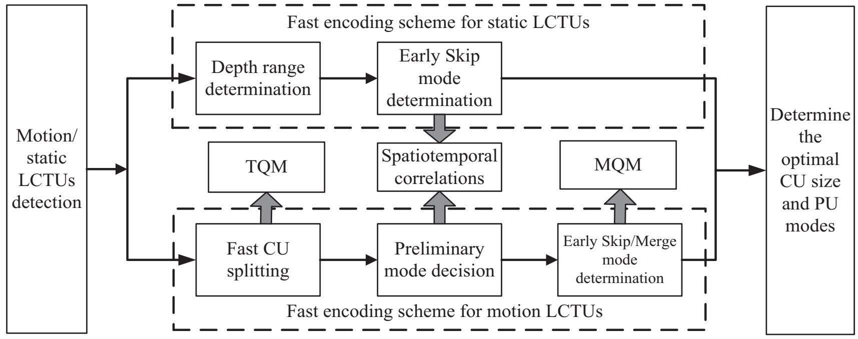


Fig. 7. Flow diagram of the overall algorithm.

3.2. Motion/static LCTUs detection and quad-tree models building

In HEVC inter prediction coding, motion and static LCTUs have different coding features because of their individual characteristics. Hence, this paper mainly explores the properties of CU splitting and mode decision of the motion and static LCTUs. The motion/static LCTUs detection algorithm adopted in this paper combines a three-frame difference method and Otsu's method [18], on the basis of which the MQM is established for each CU depth level.

The procedure of the motion/static LCTUs detection algorithm and MQM is as follows:

Step 1: Obtain the pixel values of the current B frame, forward reference frame, and backward reference frame.
Step 2: Calculate the forward frame difference $\mathbf{D_1}(x,y)$ and backward frame difference $\mathbf{D_2}(x,y)$ as follows:

$$\mathbf{D}_1(x, y) = |\mathbf{I_c}(x, y) - \mathbf{I_f}(x, y)| \tag{1}$$

$$\mathbf{D}_2(x, y) = |\mathbf{I_c}(x, y) - \mathbf{I_b}(x, y)| \tag{2}$$

where $\mathbf{I_c}(x,y)$ denotes the pixel value of position (x,y) in the current frame. In addition, $\mathbf{I_f}(x,y)$ and $\mathbf{I_b}(x,y)$ denote the pixel value of position (x,y) in the forward and backward reference frames, respectively.

Step 3: Extract the motion pixel sets of the forward and backward frame difference maps, $\mathbf{M_1}$ and $\mathbf{M_2}$, by

$$\mathbf{M}_1 = \{(x, y)|\mathbf{D}_1(x, y) \geq T_1\} \tag{3}$$

and

$$\mathbf{M}_2 = \{(x, y)|\mathbf{D}_2(x, y) \geq T_2\} \tag{4}$$

where T_1 and T_2 are the OTSU methods, and they are obtained by exhaustively searching and maximizing intra-class variance

$$T_1 = \arg\max_{0\leq i\leq 255} [\omega_{10}(i)\omega_{11}(i)(\mu_{10}(i) - \mu_{11}(i))^2] \tag{5}$$

and

$$T_2 = \arg\max_{0\leq i\leq 255} \left[\omega_{20}(i)\omega_{21}(i)(\mu_{20}(i) - \mu_{21}(i))^2\right] \tag{6}$$

where $\omega_{10}(i)$ and $\omega_{11}(i)$ are the probabilities of the motion pixels and non-motion pixels in $\mathbf{D_1}$ if the threshold is set as i, $\omega_{20}(i)$ and $\omega_{21}(i)$ are the probabilities of the motion pixels and non-motion pixels in $\mathbf{D_2}$ if the threshold is set as i, $\mu_{10}(i)$ and $\mu_{11}(i)$ represent the average frame difference of the motion pixels and non-motion pixels in $\mathbf{D_1}$ if the threshold is set as i, and $\mu_{20}(i)$ and $\mu_{21}(i)$ represent the average frame difference of the motion pixels and non-motion pixels in $\mathbf{D_2}$ if the threshold is set as i.

Step 4: Classified the LCTUs into two types, static LCTUs and motion LCTUs. Let **C** be current LCTU, it is classified according to the following rule.

$$\mathbf{C} \subset \begin{cases} Static\ \ LCTUs & if\ \ R(\mathbf{C}) \cap \mathbf{M}_1 = \varnothing \text{ and } R(\mathbf{C}) \cap \mathbf{M}_2 = \varnothing \\ Motion\ \ LCTUs & otherwise \end{cases} \tag{7}$$

Where $R(\mathbf{C})$ are the set of coordinates in **C**.

Step 5: Build an MQM for the sub-CUs of the motion LCTUs with a depth ranging from 1 to 3 by recursively employing the above motion/static detection algorithm.

An MQM is a quad-tree structure for mode prediction according to the motion/static characteristic of the CUs with various depths. In an MQM, the motion/static characteristic is judged for every CU from depths 0 to 3. The rule for building the MQM is as follows: if the current CU is classified as a motion CU, the quad-tree is split, otherwise, it is not split. The schematic diagram of the MQM is shown in Fig. 8. As shown in this figure, white blocks denote static CUs while gray blocks denote motion CUs. In Fig. 8(a), the root node of this quad-tree indicates the LCTU and each node has zero or four sub-nodes. There are four levels in this MQM, respectively corresponding to CU sizes of 64×64, 32×32, 16×16, and 8×8. The representation of the MQM in the LCTU is shown in Fig. 8(b), with flags 0 and 1 denoting static and motion CUs, respectively. Therefore, the MQM, which represents the motion and static information of various CU depth levels, is built up using the motion/static detection algorithm. According to extensive statistical experiments and analyses, this paper exploits the MQM and encoding parameters to determine Skip/Merge modes early.

Similar to an MQM, a TQM is a quad-tree structure for predicting the optimal partition of the LCTU utilizing the results of edge extraction. The building process of TQM is as follows.

Step 1: Obtain the edge map of current B frame using Sobel operator and Otsu's method.
Step 2: Build the TQM for each CU depth level using the edge map. If all pixels in current CU do not belong to edge map, the current CU is judged to be a non-edge block; otherwise, it is considered as an edge block.

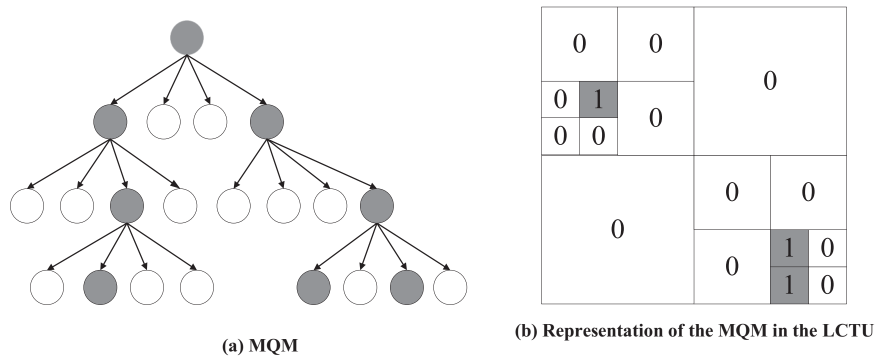


Fig. 8. MQM schematic diagram.

Because of the close relationship between the optimal CU partition in motion LCTUs and texture edge characteristics, the fast CU splitting scheme for motion LCTUs is proposed by using this model. We discuss this Scheme In detail in Section 3.4.1.

3.3. Fast encoding scheme for static LCTUs

3.3.1. Depth range determination

Using the influence of texture edge, frame type, and QP value on CU splitting, the depth range determination algorithm of static LCTUs is proposed as follows:

(a) The static LCTUs in reference B frames are encoded with fixed depths ranging from 0 to 2 while those in non-reference B frames are encoded with fixed depths ranging from 0 to 1. Because the QP value influences the CU partition, if the QP value is greater than or equal to 32, the CUs in the non-reference B frames are only encoded with depth 0. The statistical analysis is shown in Fig. 3. We can also see that the number of 64×64 CUs in the non-reference B frames are more than those in the reference B frames at the same QP level. As QP value increases, the percentage of 64×64 CUs increases while the percentage of other sizes decreases. According to Fig. 3, the hit-rate of this strategy is 92.1% on average.

(b) If the current LCTU is static or contains no edges, it is encoded with a fixed depth ranging from 0 to 1.

3.3.2. Early Skip mode determination

The CU splitting and mode selection of static LCTUs have strong temporal correlation because of the strong temporal correlations in video signal. In addition, large-size modes are usually selected as the best prediction mode in static LCTUs. Hence, early Skip mode determination mode strategy is used when the temporal collocated LCTU in the temporal reference frame is Skip mode. If the best prediction mode of the collocated LCTU in the temporal reference frame is Skip mode, the current static LCTU adopts Skip mode as the best mode and terminates CU splitting early. Otherwise, the current CU needs to be encoded by utilizing this mode, Skip, Merge, and Inter_2N × 2N.

The fast encoding scheme for static LCTUs determines most probable depth range and candidate mode sets, skipping any unnecessary encoding of small CU sizes and PU modes. Hence, it can effectively speed up the inter-coding process on the premise of negligible RD performance loss.

3.4. Fast encoding scheme for motion LCTUs

3.4.1. Fast CU splitting

The optimal CU size of motion LCTUs is closely related to their texture edge characteristics. Hence, the proposed algorithm exploits the relationship between CU partition size and texture edge characteristics and builds the TQM accordingly. Fast CU splitting for motion LCTUs is performed using the TQM and temporal correlation discussed below.

The specific scheme proceeds as follows:

(a) Build the TQM for each CU depth level using the edge-map.

(b) Use temporal correlation to calculate d_{ref}, the reference depth of the current CU, as

$$d_{ref} = (d_f + d_b + 1)/2 \tag{8}$$

where d_f and d_b are the optimal depth of the collocated CU in the forward and backward reference frames, respectively. Note that the reference frames are determined by the hierarchical B structure.

(c) Calculate the final coding depth of the current motion CU as

$$d_{move} = \max(d_{ref}, d_t) \tag{9}$$

where d_{move} is the final coding depth of the current motion CU and d_t is the coding depth determined by the TQM.

3.4.2. Preliminary mode decision

There is a strong spatial correlation in video frames, the optimal mode between adjacent CUs among video frames is also strongly spatially correlated. Based on this principle, a preliminary mode decision can be made by eliminating nonexistent modes in spatially adjacent CUs. This strategy is only used for depth levels 0 and 1. This is because motion CUs generally choose a small size (namely, 16×16 or 8×8) as the best size in terms of RDO. Therefore, preliminary mode decisions for large sizes do not lead to larger RD performance loss.

The preliminary mode decision for motion CUs is as follows. First, the three frequent modes, Inter_2N × 2N, Skip, and Merge, are joined in candidate mode set for motion CUs. If the best mode of the spatially adjacent CUs is M, M is added to the candidate mode set. After traversing all spatially adjacent CUs of the current motion CU, we obtain the final candidate mode set. Finally, the current CU just calculates the modes in the candidate mode set to determine the preliminary mode decision.

3.4.3. Early Skip/Merge mode determination

Motion LCTUs detected by the motion/static detection algorithm contain many static or slow motion sub-CUs that are likely to be encoded with Skip or Merge mode. As a result, we propose early Skip/Merge mode determination as follows. The CUs with depth levels from 1 to 3 in motion LCTUs are further classified into motion CUs or static CUs based on the MQM. If the current CU satisfies the following condition:

Table 1
Percentages of Skip/Merge mode that satisfy (10) for different sequences (%).

QP	a	b	c	d	e	Average
22	92.5	86.3	90.1	85.5	98.7	90.6
27	96.0	91.7	93.4	90.8	99.6	94.3
32	97.9	94.9	95.4	94.1	99.8	96.4
37	98.4	96.9	97.0	96.4	99.9	97.7
Average	96.2	92.5	94.0	91.7	99.5	94.8

a: Traffic, b: Kimono, c: PartyScene, d: BlowingBubbles, e: Johnny [16]

$$\begin{cases} \text{Condition 1: static CUs, for depth} = 1, 2, 3 \\ \text{Condition 2: CBF} = 0 \ \ or \ \ \text{MV} = (0, 0) \end{cases} \tag{10}$$

we determine that the best mode of the current CU is Skip or Merge mode, and thus all other possible modes are skipped.

In order to validate the encoding hit-rate under the conditions in (10), this study analyzed the prediction mode distribution that satisfies (10) for five video sequences that have different spatial resolutions and texture properties. The encoding hit-rate for different sequences is shown in Table 1.

As shown in Table 1, the percentage of sub-CUs of motion LCTUs for which the best mode is Skip or Merge is 94.8% on average. Therefore, the sub-CUs of motion LCTUs that satisfy (10) can choose Skip or Merge mode as the best mode, thus skipping the ME process of the other inter modes and the prediction of the intra modes.

4. Experimental results and analyses

To validate the efficacy of the proposed fast inter coding algorithm, we incorporated the proposed algorithm into HM-12.0 and HM-16.9 to test the RD performance and coding complexity. The test platform used for the experiments was a 64-bit Microsoft Windows 7 operating system running on a PC with Intel Core i5 CPU at 3.3 GHZ and 8.0 GB RAM. The RA-Main profile was used for the experiments with four QPs, 22, 27, 32, and 37. The number of pictures to be encoded was full frames and the size of the GOP (Group of Pictures) was 8. The RD performance of the proposed algorithm is measured by the Bjontegaard delta bit rate (BDBR) [19]. Thirteen test sequences in common test conditions [16] were used for the experiments. To validate the effectiveness of the proposed algorithm, the original HM-12.0 and HM-16.9, and the state-of-the-art method Ahn's algorithm [11] were used for comparison. Besides, we present the performance of Schemes I and II to further clarify the contribution of proposed algorithm.

Table 2 lists the experimental results of various encoding schemes where BDBR_i (i={Scheme I, Scheme II, Proposed, Ahn}) is the BDBR of the ith scheme, and time saving ΔT_i is defined as

$$\Delta T_i = \frac{\text{Time}_{original} - \text{Time}_i}{\text{Time}_{original}} \times 100\%$$
$$i = \left\{ \text{Scheme I, Scheme II, Proposed, Ahn} \right\} \tag{9}$$

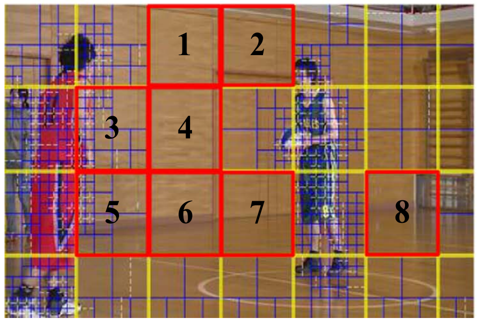
where $\text{Time}_{original}$ and Time_i indicate the encoding time of the original HM scheme and the ith scheme. Schemes I and II can reduce encoding time by 26.6% and 35.0% on average, respectively, with negligible RD performance loss. The proposed algorithm, fused Schemes I and II, reduces the encoding time by 47.5% on average with a BDBR increase of only 1.6% (with a maximum of 63.6% time reduction for sequence “FourPeople” and a minimum of 34.5% for sequence “BQSquare”).

The proposed algorithm outperforms Ahns algorithm with 1.1% more encoding time saving with comparable RD performance. The improvement of speedup performance is contributed by jointly utilizing the MQM and TQM, spatiotemporal correlations, and statistical properties. First, there are quite a few static blocks in video sequences, and Scheme I takes texture edge, frame type, QP value, and temporal correlations into consideration. Hence, the encoding hit-rate of the depth range determination and mode selection of static LCTUs in Scheme I is extremely high, this scheme can reduce complexity by 26.6% on average with negligible amounts of BDBR increase. Furthermore, in Scheme II, fast CU splitting for motion LCTUs is performed using the TQM and temporal correlation of CU size, preliminary mode decision and early Skip/Merge mode determination are performed by utilizing the MQM and spatial correlations. This scheme makes full use of the motion/static and statistical characteristics of each CU depth level, thus it can save a notable amount of encoding time by 35.0% while maintaining good RD performance. Lastly, the proposed algorithm, which combines Schemes I and II, can save notably more encoding time with acceptable BDBR increase. It can save much more encoding time than the individual Schemes I and II because these two schemes perform fast encoding from different perspectives and predict accurately. It is interesting to note that, for the test sequences of Class E, the proposed overall algorithm can reduce the total encoding time by the substantial amount of 62.0% with negligible RD performance loss.

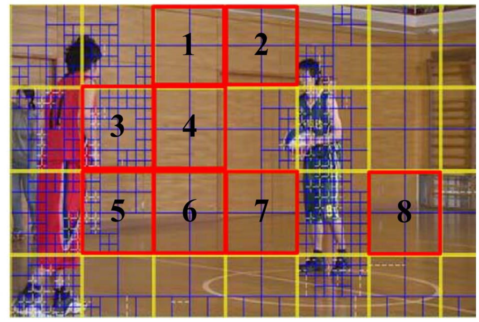
Ahn's algorithm reduces encoding complexity by 47.8% on

Table 2
Experimental results of each scheme compared with HM (%).

Sequence		Scheme I (HM-16.9)		Scheme II (HM-16.9)		Proposed algorithm (HM-16.9)		Proposed algorithm (HM-12.0)		Ahn's algorithm (HM-12.0)	
		$\text{BDBR}_{Scheme\ I}$	$\Delta T_{Scheme\ I}$	$\text{BDBR}_{Scheme\ II}$	$\Delta T_{Scheme\ II}$	$\text{BDBR}_{Proposed}$	$\Delta T_{Proposed}$	$\text{BDBR}_{Proposed}$	$\Delta T_{Proposed}$	BDBR_{Ahn}	ΔT_{Ahn}
Class A	PeopleOnStreet	0.1	16.2	0.8	36.9	2.1	49.9	2.3	51.6	0.9	26.9
Class B	ParkScene	0.4	18.2	0.6	34.7	1.4	46.6	1.4	48.2	1.2	52.6
	Cactus	0.3	33.5	1.0	38.0	1.5	55.9	1.6	54.6	2.8	56.8
	BasketballDrive	1.0	32.5	1.5	45.0	1.4	52.6	1.8	55.4	2.0	50.9
	BQTerrace	0.8	29.2	0.4	37.1	0.8	50.6	0.8	48.9	1.6	54.5
Class C	BasketballDrill	0.6	27.5	1.1	33.9	1.6	47.7	1.5	48.7	1.9	45.2
	BQMall	0.7	20.9	0.8	30.4	1.5	40.3	1.5	46.8	2.2	48.6
	RaceHorsesC	0.3	21.5	0.7	31.1	1.8	43.6	1.9	44.2	2.2	33.9
Class D	BasketballPass	0.6	18.2	0.6	32.3	1.8	35.5	2.9	45.2	1.5	33.6
	BQSquare	1.3	17.7	0.3	23.3	2.0	34.5	1.0	34.8	0.6	45.1
	RaceHorses	0.2	15.7	0.7	25.9	2.5	36.0	2.6	38.6	1.1	26.6
Class E	FourPeople	0.1	47.1	1.4	39.8	1.1	63.6	0.9	62.7	1.7	74.1
	KristenAndSara	1.3	47.7	1.2	47.1	1.1	60.4	0.8	56.4	1.2	73.1
Average		0.6	26.6	0.9	35.0	1.6	47.5	1.6	48.9	1.6	47.8

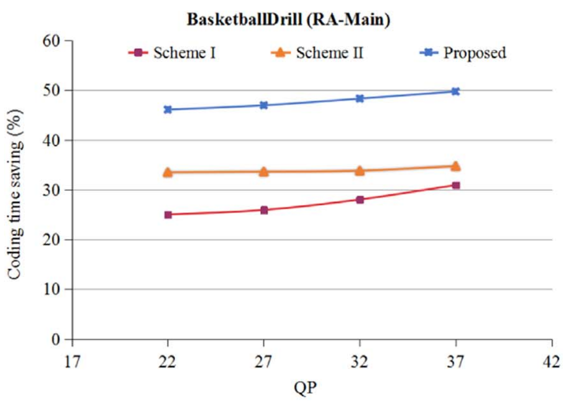


(a) The LCTU partition of 16st frame under original HM-16.9 scheme (QP=27)

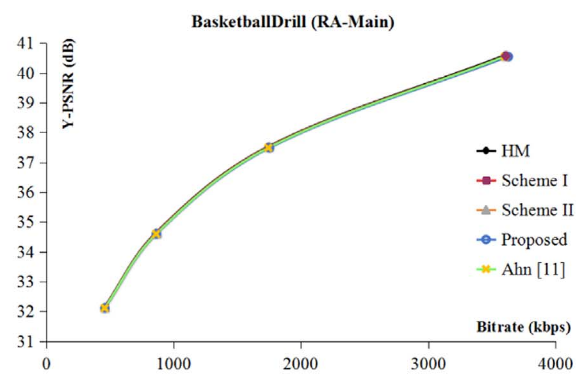


(b) The LCTU partition of 16st frame under proposed algorithm (QP=27)

Fig. 9. The LCTU partition for the "BasketballPass" sequence under different encoding schemes.



(a) Coding time saving comparison



(b) RD performance comparison

Fig. 10. Comparison of RD performance and coding time saving of each scheme.

average with a BDBR increase of 1.6% (sequence "Fourpeople" has a maximum of 74.1% time reduction, and sequence "RaceHorses" has a minimum 26.6%). As far as the average time saving is concerned, the proposed algorithm slightly outperforms Ahn's algorithm, with 1.1% more encoding time saving and comparable RD performance. Compared with Ahn's algorithm, our algorithm is efficient for the sequences with slow motion scene because high prediction of CU splitting and mode selection can be obtained. Hence, for the sequences "Cactus", "BasketballDrill", "BQTerrace", and "BQMall", our proposed algorithm outperforms than Ahn's algorithm in terms of RD performance. However, for the sequences with dramatic scene switching, e.g., "BasketballPass" and "RaceHorses", the proposed algorithm slightly decreases the RD performance because the prediction of the quad-tree models is not precise enough. Fig. 9 illustrates the CU splitting difference between the proposed algorithm and original encoding scheme. In the eight collocated LCTUs, the proposed algorithm has more elaborate CU partition. Consequently, both the bitrate and BDBR are increased.

Fig. 10 shows the RD performance and coding time saving of each scheme for the "BasketballDrill" sequence under the RA-Main profile. As shown in Fig. 10, the proposed combined algorithm can save more calculation time and each scheme can maintain a good RD performance. In Fig. 10(a), we can also see that the coding time saving of the proposed algorithm increases as QP increases. As shown in Fig. 10(b), there is almost no loss of RD performance as the bit rate increases. In conclusion, the proposed algorithm can effectively reduce encoding complexity for HEVC inter coding with acceptable RD performance loss.

5. Conclusion

In order to reduce the inter encoding complexity of HEVC encoders, this paper proposed a fast inter coding algorithm based on texture and motion quad-tree models. The proposed algorithm includes two fast encoding schemes, one for static LCTUs (Scheme I) and the other for motion LCTUs (Scheme II). Scheme I determines the depth range and candidate mode set of static LCTUs by fully utilizing statistical properties and temporal correlations. In Scheme II, fast CU splitting for motion LCTUs is based on the TQM and temporal correlation of CU size while preliminary mode decisions and early Skip/Merge mode determinations are made by the MQM and spatial correlation so that unnecessary modes may be skipped. The experimental results show that the proposed combined algorithm can save 47.5% encoding time with only a 1.6% BDBR increase on average. Our algorithm also slightly outperforms Ahn's algorithm in terms of average time saving, and has higher efficiency for the sequences with slow motion scene. In the future work, we will improve the encoding speed and enhance the RD performance of the sequence with drastic motion scene.

Acknowledgment

This work was supported by the Natural Science Foundation of China (61271270, 61111140392, U1301257), National High-tech R and D Program of China (863 Program, 2015AA015901), Natural Science Foundation of Zhejiang Province (LY15F010005, LY16F010002) and Natural Science Foundation of Ningbo (2015A610127, 2015A610124). It is also sponsored by K.C. Wong Magna Fund in Ningbo University.

References

[1] T. Wiegand, J.R. Ohm, G.J. Sullivan, et al., Special section on the joint call for proposals on high efficiency video coding (HEVC) Standardization, IEEE Trans. Circuits Syst. Video Technol. 20 (12) (2010) 1661–1666.
[2] T. Wiegand, G.J. Sullivan, G. Bjontrgaard, et al., Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2012) 560–576.
[3] J. Vanne, M. Viitanen, D.H. Timo, et al., Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1885–1898.
[4] G.J. Sullivan, J. Ohm, W.J. Han, et al., Overview of the high efficiency video coding (HEVC) standard, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1649–1668.
[5] J.R. Ohm, G.J. Sullivan, H. Schwarzohm, et al., Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC), IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1669–1684.
[6] R.H. Gweon, J.L. Lee, J. Lim, Early termination of CU encoding to reduce HEVC complexity, in: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-F045, 6th Meeting, Torino, IT, July 2011.
[7] K. Choi, S.H. Park, E.S. Jang, Coding tree pruning based CU early termination, in: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-F092, 6th Meeting, Torino, IT, July 2011.
[8] J. Yang, J. Kim, K. Won, et al., Early SKIP detection for HEVC, in: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-G543, 7th Meeting, Gevena, CH, November 2011.
[9] L.Q. Shen, Z.Y. Zhang, Z. Liu, Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations, IEEE Trans. Circuits Syst. Video Technol. 24 (10) (2014) 1709–1722.
[10] H. Lee, H.J. Shim, Y. Park, et al., Early skip mode decision for HEVC encoder with emphasis on coding quality, IEEE Trans. Broadcast. 61 (3) (2015) 388–397.
[11] S. Ahn, B. Lee, M. Kim, A novel fast CU encoding scheme based on spatio-temporal encoding parameters for HEVC inter coding, IEEE Trans. Circuits Syst. Video Technol. 25 (3) (2015) 422–435.
[12] S.Y. Jou, S.J. Chang, T.S. Chang, Fast motion estimation algorithm and design for real time QFHD high efficiency video coding, IEEE Trans. Circuits Syst. Video Technol. 25 (9) (2015) 1533–1544.
[13] A. Lee, D.S. Jun, S.C. Jin, Fast motion estimation using priority-based inter-prediction mode decision method in high efficiency video coding, J. Real.-Time Image Process (2015) 1–9.
[14] C.G. Yan, Y.D. Zhang, J.Z. Xu, et al., A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors, IEEE Signal Process. Lett. 21 (5) (2014) 573–576.
[15] C.G. Yan, Y.D. Zhang, J.Z. Xu, et al., Efficient parallel framework for HEVC motion estimation on many-core processors, IEEE Trans. Circuits Syst. Video Technol. 24 (12) (2014) 2077–2089.
[16] F. Bossen, Common test conditions and software reference configurations, in: Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Document JCTVC-L1100, 12th Meeting, Geneva, CH, January 2013.
[17] F. Bossen, B. Bross, K. Sühring, HEVC complexity and implementation analysis, IEEE Trans. Circuits Syst. Video Technol. 22 (12) (2012) 1685–1696.
[18] N. Otsu, A. Threshold, Selection Method from Gray-Level Histograms[J], IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[19] G. Bjontegaard, Calculation of average PSNR differences between RD-Curves, in: Proceedings of the 13th VCEG-M33 Meeting, Austin, TX, April 2001.