

A Fast TU Size Decision Method for HEVC RQT Coding

Jinfu Wu, Baolong Guo, Yunyi Yan, Jie Hou, and Dan Zhao

School of Aerospace Science and Technology, Xidian University
Xi'an, Shaanxi 710071, China

[e-mail: wjf505@gmail.com, {blguo, yyyan}@xidian.edu.cn,
jie.hou.xdu@gmail.com, fengjiran@foxmail.com]

*Corresponding author: Yunyi Yan

*Received December 28, 2014; revised March 10, 2015; revised April 20, 2015; revised May 14, 2015;
accepted May 26, 2015; published June 30, 2015*

Abstract

The emerging high efficiency video coding (HEVC) standard adopts the quadtree-structured transform unit (TU) in the residual quadtree (RQT) coding. Each TU allows to be split into four equal sub-TUs recursively. The RQT coding is performed for all the possible transform depth levels to achieve the highest coding efficiency, but it requires a very high computational complexity for HEVC encoders. In order to reduce the computational complexity requested by the RQT coding, in this paper, we propose a fast TU size decision method incorporating an adaptive maximum transform depth determination (AMTD) algorithm and a full check skipping - early termination (FCS-ET) algorithm. Because the optimal transform depth level is highly content-dependent, it is not necessary to perform the RQT coding at all transform depth levels. By the AMTD algorithm, the maximum transform depth level is determined for current treeblock to skip those transform depth levels rarely used by its spatially adjacent treeblocks. Additionally, the FCS-ET algorithm is introduced to exploit the correlations of transform depth level between four sub-CUs generated by one coding unit (CU) quadtree partitioning. Experimental results demonstrate that the proposed overall algorithm significantly reduces on average 21% computational complexity while maintaining almost the same rate distortion (RD) performance as the HEVC test model reference software, HM 13.0.

Keywords: TU size decision, HEVC/H.265, RQT coding, video compression

1. Introduction

The High Efficiency Video Coding (HEVC) standard is the most recent joint video project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations, working together in a partnership known as the Joint Collaborative Team on Video Coding (JCT-VC) [1]. HEVC significantly outperforms previous standards such as MPEG-2, MPEG-4 Part 2 and H.264/AVC in terms of coding efficiency, and achieves half reduction of the bitrate needed by the widely used H.264/AVC while maintaining equivalent visual quality [2, 3].

To obtain higher coding efficiency, a large number of new techniques are adopted in HEVC, such as the hierarchical quadtree structure of motion compensation, large coding treeblock, coding unit (CU), prediction unit (PU) and transform unit (TU) [4]. A treeblock consists of 64×64 luma samples together with two corresponding blocks of chroma samples. To some extent, treeblock in HEVC is broadly analogous to macroblock (MB) in H.264/AVC. CU is the basic unit for intra/inter prediction, which allows to be subdivided into four equally sized sub-CUs recursively. CU is always square and of size from the size of treeblock down to 8×8 . PU, which covered by the prediction residual block, can be recursively partitioned into multiple square TUs in a way known as the residual quadtree (RQT) coding. TU supports transform block with size from 4×4 to 32×32 [1-4]. Both CUs and TUs can be recursively subdivided into quadrants as illustrated in Fig. 1, where solid lines indicate CU boundaries and dotted lines indicate TU boundaries.

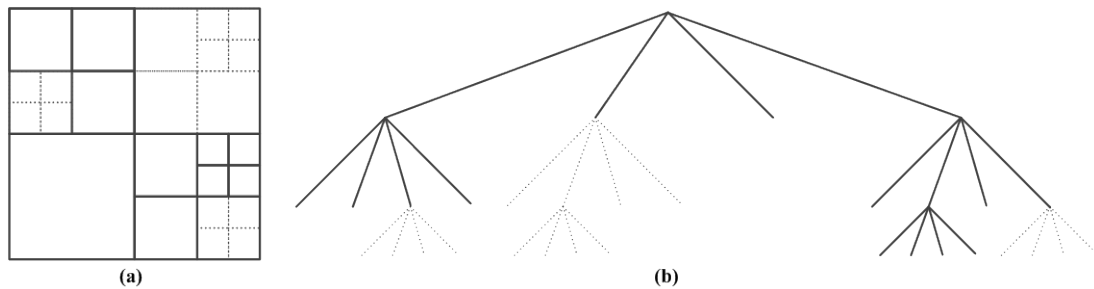


Fig. 1. Subdivision of a treeblock into CUs [and TUs]. Solid lines indicate CU boundaries and dotted lines indicate TU boundaries. (a) Treeblock with its partitioning. (b) Corresponding quadtree.

According to the HEVC standard [1], an image is first divided into a number of treeblocks. Each treeblock is then further partitioned into multiple CUs. CU serves as the root for the TU quadtree structure as shown in Fig. 1. In the TU quadtree structure, current TU may be identical to the CU it belongs to, otherwise current TU will be split into four smaller TUs. It is referred to as the residual quadtree (RQT) coding which takes a combination of operations: transform, quantization, inverse quantization and inverse transform to determine the optimal TU size in a brute-force fashion [1, 5-7]. The optimal TU size, i.e., the one with the least rate distortion (RD) cost, is determined according to Lagrangian multiplier (more details can be found in [3, 8]). RD cost for each TU size decision is defined as follows:

$$RDcost_x = B_x + \lambda_x \cdot SSE_x \quad (1)$$

where X stands for the TU size to be check, B_x represents the bit cost considered for the TU size decision, SSE_x (Sum Squared Error) represents the difference between the original residual image covered by the tested TU and its reconstruction, λ_x is the Lagrangian multiplier determined by the content of the tested TU.

However, this “try all and select the best” method adopted by the RQT coding will result in high computational complexity and limit real-time HEVC encoders in applications. Therefore, fast algorithms which are expected to reduce the computational complexity requested by the RQT coding without sacrificing too much coding efficiency, are very desirable for real-time implementation of HEVC encoders.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review of fast algorithms for HEVC encoders. Section 3 describes the proposed fast TU size decision method for the RQT coding in HEVC, which incorporates an adaptive maximum transform depth determination (AMTD) algorithm and a full check skipping - early termination (FCS-ET) algorithm. Experiments are carried out and analyzed in Section 4 to demonstrate the effectiveness of the proposed algorithms. Section 5 concludes the paper.

2. Related Work

Recently, many fast algorithms have been proposed for HEVC encoders to reduce the computational complexity. In general, these algorithms could be categorized into three main classes: fast CU size decision, fast PU mode decision and fast TU size decision. They are briefly summarized as follows.

Fast Coding Unit Size Decision: Based on the observation that current CU will not be subdivided if its coding mode is SKIP mode, Choi and Jang proposed a tree pruning algorithm for fast CU size decision [9]. It was reported that a 40% reduction in encoding time can be achieved on HM 3.0. However, this fast CU size decision algorithm is applied to inter prediction rather than intra prediction.

Shen *et al.* [10] proposed a fast CU size decision scheme based on the Bayesian decision rule. The CU size decision was made on the Bayesian risk, which can be calculated from the Lagrangian cost, the class-conditional probability density functions and priori probabilities. Random access (RA) and low-delay B (LB) configurations were used for simulations and on average 41.4% encoding time reduction was reported. But the decision on TU size and mode selection are not investigated.

A gradient based fast intra mode decision was proposed by Jiang [11], where gradient directions and histogram were derived for fast CU size decision. Based on the distribution of gradient histogram, only a small part of candidate modes were chosen for rough mode decision (RMD) and rate distortion optimization (RDO) process. This approach can reduce on average 20% encoding time with negligible loss of coding efficiency on HM 4.0. Whereas this fast CU size decision is just for intra mode decision.

Shen *et al.* [12, 13] proposed an adaptive CU depth range determination method and several early termination schemes to speed up the CU size decision based on the spatial correlations between neighboring CUs. With these algorithms, 42% and 41% encoding time was reduced under RA and LB configurations on HM 2.0, respectively. However, those techniques such as early termination of motion estimation are applied to inter mode decision only and are not fit for intra mode decision.

Fast Prediction Unit Mode Decision: Zhao *et al.* [14] have studied the impact of the number of PU mode candidates after the RMD process, and proposed a fast PU mode decision algorithm for intra prediction by reducing the number of PU mode candidates. It was reported that on average 20% and 28% encoding time reduction can be achieved under High Efficiency (HE) and Low Complexity (LC) test conditions, respectively, with almost the same coding efficiency as HM 1.0. It is noted that HE and LC test conditions are used in the early phase of HEVC development and have been merged since HM 6.0.

Several fast intra prediction mode decisions have been introduced by Zhang and Ma [15, 16]. In their works, three methods were proposed to speed up the PU mode decision: a modified Hadamard transform, a novel progressive rough mode search and an early termination method for rate distortion optimization quantization (RDOQ) process. It was reported 2.5x speedup with 1.0% BD-Rate increase on HM 10.0. However, the correlations between neighboring TUs have not yet been exploited.

Fast Transform Unit Size Decision: The RQT coding in HEVC is applied to improve the coding efficiency, but it demands significant computational overhead [2, 3, 5-8]. Hence, Tan *et al.* proposed several fast encoding schemes for both intra prediction and inter prediction for TU size decision-making in the RQT coding [17]. It was reported that for AI case, the fast RQT coding algorithm saved 13% encoding time with 0.1% BD-Rate increase. For RA and LB cases, up to 9% encoding time can be reduced at the expense of 0.3% BD-Rate increase on HM 2.0.

Another fast RQT coding scheme was proposed by Teng *et al.* [18], where the original depth-first TU size decision was replaced by a Merge-and-Split decision for the RQT coding. The Merge-and-Split decision process was terminated and no further TU splitting was performed when current TU was a zero-block. It reported almost 2x speedup for RA case under HE configuration on HM 2.0, with about 0.3% BD-Rate increase.

Kiho Choi *et al.* [19] exploited the relationship between the determined TU size and the number of nonzero DCT coefficients to determine the TU size at an early stage. The total misprediction ratio for the given NNZ with a threshold 3 was 1.26%, which implied that there existed negligible coding efficiency compared to the original HEVC encoder. It showed 0.6% BD-Rate increase as well as averaged 60% computational complexity reduction for RA case under HE configuration on HM 3.0.

Furthermore, Zhang and Zhao [20] have developed an adaptive RQT coding algorithm for inter prediction by restricting the smaller transform depth level for larger CU size and vice versa, based on the observations from the RQT coding. It reported 0.7% BD-Rate increase with 7.2%~21% computational complexity reduction under HE and LC configuration on HM 4.0, respectively. However, the proposed algorithm is for inter prediction only and intra prediction is not concerned.

A piece of our earlier work presented a fast RQT coding algorithm based on the experimental observations [21]. When current CU was split into four quadrants, the RQT coding of upper-left sub-CU was performed first, followed by the individual RQT coding of remaining three sub-CUs with predetermining their smallest TU size as that of upper-left sub-CU. Therefore, a lot of TU size decisions can be skipped for the RQT coding. This fast RQT coding algorithm is updated to reduce more TU size decisions in Section 3.

The aforementioned algorithms are well developed to reduce the computational complexity for HEVC encoders. However, the TU size correlations between neighboring CUs are not fully studied. In order to reduce the computational complexity requested by the RQT

coding, in this paper, we propose an effective TU size decision method incorporating an adaptive maximum transform depth determination (AMTD) algorithm and a full check skipping - early termination (FCS-ET) algorithm. Considering that the optimal transform depth level, namely TU size, for a certain treeblock is highly content-dependent, it is not necessary to perform the RQT coding at all transform depth levels for all treeblocks in a given video sequence. Therefore, the AMTD algorithm is introduced to skip those transform depth levels rarely used in spatially adjacent treeblocks for current treeblock according to their spatial correlations. Meanwhile, there exist spatial and temporal correlations between four sub-CUs generated by one CU quadtree partitioning. By the FCS-ET algorithm, the unnecessary RQT coding on small CU sizes can be skipped to reduce the encoding time. Experimental results demonstrate that the proposed overall algorithm can significantly reduce the computational complexity while maintaining almost the same coding efficiency as the HEVC test model reference software, HM 13.0.

3. Overview of Fast TU Size Decision Method for HEVC

3.1 Adaptive Maximum Transform Depth Determination (AMTD)

In HEVC the transform depth level of the treeblock is within a fixed range (i.e., 0~3), that is, the available TU sizes are from 32×32 down to 4×4 . Current TU is implicitly split when its size is larger than the maximum TU size, and it will be not split when the TU splitting would result in a TU size smaller than the minimum TU size. By allowing different TU sizes, the RQT coding enables the adaptation of the transform to the varying space-frequency characteristics of the residual signal. Larger TU sizes, which have larger spatial support, provide better frequency resolution, while smaller TU sizes, which have smaller spatial support, provide better spatial resolution [1-5, 7, 8].

It becomes evident that the transform depth level range of current treeblock should be adaptively determined based on its motion and texture properties, so the exhaustive TU size decision adopted by the original RQT coding is inefficient. The fast RQT coding algorithms [17-20] mentioned in Section 2 simply utilize the characteristics of the RQT coding, such as the number of nonzero transform coefficients and the obvious correlations between TU size and CU size. However, the correlations of transform depth level between current treeblock and its spatially adjacent treeblocks are not fully utilized.

Natural video sequences have inherently strong spatial and temporal correlations, especially in the homogeneous regions. The fact that there exist a mass of homogeneous regions in the natural video sequences means that the texture between adjacent treeblocks are similar. So the optimal CU depth level and transform depth level of current treeblock is the same or very close to those of its spatially coded adjacent treeblocks due to their high spatial correlations [22]. Specifically, the optimal transform depth level of current treeblock $TrDepth_{pred}$ can be predicted from the transform depth levels of its spatially coded adjacent treeblocks (L , U , and $L-U$ treeblocks in Fig. 2) as follows:

$$TrDepth_{pred} = \sum_{i=1}^N \omega_i \cdot \delta_i \quad (2)$$

where N is the number of spatially coded adjacent treeblocks of current treeblock equal to 3 in our experiments, ω_i is the predetermined correlation weight for i th neighboring treeblock

and δ_i is the transform depth level of i th neighboring treeblock. The correlation weights are normalized to have $\sum_{i=1}^3 \omega_i = 1$.

Those three coded treeblocks denoted as L , U , and $L-U$ treeblocks in **Fig. 2** are chosen for current treeblock for the prediction function (2) due to their higher spatial correlations. And the top-right treeblock and the colocated treeblock of current treeblock are excluded in our experiments to provide a desirable balance between computational complexity and coding efficiency. In general, the spatial correlation between treeblocks is higher as they are closer. It is obvious that the upper and left treeblocks are nearer (in terms of Euclidean distance) to current treeblock relative to the upper-left treeblock, that is, the spatially adjacent treeblocks in the horizontal and vertical directions have a higher correlation with current treeblock than the treeblock in the diagonal direction. Due to the diversity of natural videos, it is impractical to assign different correlation weights for upper and left treeblocks for all video sequences. Therefore, a large correlation weight is assigned for both the upper treeblock and the left treeblock, while a small one for the upper-left treeblock in the prediction function (2). The experiments to choose the appropriate weights for these three coded treeblocks are presented in the following section.

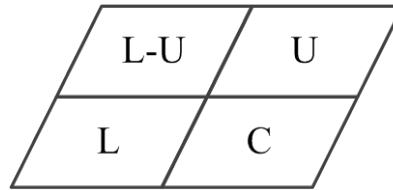


Fig. 2. Spatial relationship of treeblocks. C : current treeblock; L : left treeblock; U : upper treeblock; $L-U$: upper-left treeblock.

According to the predicted value of the optimal transform depth level, each treeblock is divided into one of three types (G_1 , G_2 , G_3) as follows:

- (1) If $0 \leq TrDepth_{pred} \leq 0.5$, most of spatially coded adjacent treeblocks of current treeblock tend to choose a small transform depth level “0”. Current treeblock is located in the moderate motion region and classified as type G_1 ;
- (2) If $0.5 < TrDepth_{pred} \leq 1.5$, most of spatially coded adjacent treeblocks of current treeblock tend to choose transform depth level “1”. Current treeblock is classified as type G_2 ;
- (3) If $1.5 < TrDepth_{pred} \leq 3$, most of spatially coded adjacent treeblocks of current treeblock tend to a large choose transform depth level “2” and “3”. Current treeblock is located in the fast motion region and classified as type G_3 .

For type G_1 treeblock, the area its coded spatially adjacent treeblocks cover contains motionless or slow-motion content. So it is good enough to perform the RQT coding for type G_1 treeblock on large TU size, i.e., 16×16 and 32×32 . For type G_3 treeblock, most of its spatially coded adjacent treeblocks are located in the fast motion region. As a result, the RQT coding remains unchanged as in the original HEVC encoder for type G_3 treeblock to achieve high coding efficiency. **Table 1** summarizes the candidate transform depth levels to be tested in the RQT coding for each type of treeblock. As shown in **Table 1**, type G_1 and G_2 treeblocks

can skip 1~2 unnecessary transform depth levels. The maximum transform depth level for type G_3 treeblock is 3, consequently, the original RQT coding is used.

Table 1. Candidate transform depth levels for each treeblock type

Treeblock type	Candidate transform depth levels	Maximum transform depth level
G_1	0, 1	1
G_2	0, 1, 2	2
G_3	0, 1, 2, 3	3

Extensive experiments are conducted to verify the legitimacy of the proposed AMTD algorithm. Six video sequences are selected: “BasketballDrive” and “Cactus” (1920×1080), “Johnny” and “KristenAndSara” (1280×720), “BQMall” and “PartyScene” (832×480). Among these video sequences, “BasketballDrive” and “Cactus” are with a large global motion or a large local motion; “Johnny”, “KristenAndSara”, “BQMall” and “PartyScene” are with a medium local motion. Test conditions are listed as follows: each video sequence is encoded under both RA and LB configurations; quantization parameter (QP) is set to 22, 27, 32, 37, which are indicated in [23]. RDOQ is enabled.

The prediction accuracy of the transform depth level determined by the AMTD algorithm using different correlation weight sets are listed in the columns 2~4 of Table 2. It can be seen that more than 95% treeblocks will fall in their predetermined transform depth level range. The reason is that the dominant treeblocks with TU size 8×8 and 4×4 will always be divided into type G_2 or G_3 treeblock according to prediction function (2). For type G_2 or G_3 treeblock, its RQT coding is the same as the default RQT coding in HEVC except that type G_2 treeblock skips checking transform depth level 3 in the RQT coding, which ensures the high accuracy of prediction function (2). It should be noted that the accuracy of prediction function (2) is higher as the resolution of video sequences increases, because there exist much more homogeneous regions, thus much more spatial correlation to be used for prediction in the video sequence with a higher resolution. An accuracy of 97% can even be achieved using the correlation weight set $\{0.4, 0.4, 0.2\}$. Note that the difference of prediction accuracy using different correlation weight sets is small, for which an explanation is given in the columns 5~7 of Table 2, where the prediction accuracy of the transform depth level for each type of treeblock is presented. The prediction accuracy of type G_2 and G_3 treeblocks is nearly 100% using the correlation weight set $\{0.4, 0.4, 0.2\}$, and it holds for other correlation weight sets such as $\{0.35, 0.35, 0.3\}$ and $\{0.45, 0.45, 0.1\}$. As a consequence, the correlation weight set $\{0.4, 0.4, 0.2\}$ is chosen for the AMTD algorithm. From Table 2 we can anticipate that by the AMTD algorithm, many TU size decisions can be skipped without sacrificing too much coding efficiency. The anticipation here will be further verified in our following experiments.

Table 2. Statistical analysis of transform depth level distribution accuracy (%)

Sequences	Accuracy of transform depth level distribution of three different correlation weight sets			Accuracy of transform depth level distribution for three treeblock types using $\{0.4, 0.4, 0.2\}$		
	$\{0.35, 0.35, 0.3\}$	$\{0.4, 0.4, 0.2\}$	$\{0.45, 0.45, 0.1\}$	$G_1, [0, 1]$	$G_2, [0, 1, 2]$	$G_3, [0, 1, 2, 3]$
BasketballDrive	96.5	97.1	97.4	95.5	100	100
Cactus	98.4	98.5	97.5	96.1	100	100
Johnny	97.2	99.2	98.1	99.1	100	100

KristenAndSara	96.8	98.6	95.7	98.6	100	100
BQMall	95.5	94.5	93.3	93.3	100	100
PartyScene	94.4	95.9	92.9	95.5	100	100
Average	96.5	97.3	95.8	96.4	100	100

3.2 Full Check Skipping and Early Termination (FCS-ET) Algorithm

As previously mentioned, the original RQT coding in HEVC is conducted to determine the optimal TU size in a brute-force fashion for current CU. At each transform depth level (i.e., TU size), full check process is performed first to obtain the RD cost, i.e., $RD_{cost_{full}}$, for encoding the non-splitting TU. Then $RD_{cost_{separate}}$ is calculated, which is the total RD cost needed for encoding four smaller TUs resulted from the TU splitting. When $RD_{cost_{full}} < RD_{cost_{separate}}$, the size of non-splitting TU is chosen as the optimal TU size for current CU. Otherwise, the size of smaller TU is chosen as the optimal TU size for current CU. Hence, high computational complexity is introduced by the original RQT coding [5].

In our previous work [21], an early determination algorithm for the TU size decision was proposed based on the experimental observations. When current CU was split into four quadrants, the RQT coding of upper-left sub-CU was performed first, followed by the individual RQT coding of remaining three sub-CUs with predetermining their smallest TU size as that of upper-left sub-CU. Once the TU size of current sub-CU exceeds the predetermined smallest TU size, the RQT coding is early terminated and no more RD cost evaluations are further conducted.

Due to the strong correlations between four sub-CUs generated by one CU quadtree partitioning, the optimal transform depth level of current sub-CU is the same or very close to that of its spatially adjacent sub-CUs. By taking full advantage of these transform depth level correlations, we propose a full check skipping - early termination (FCS-ET) algorithm, which combines our previous early termination algorithm with a full check skipping algorithm.

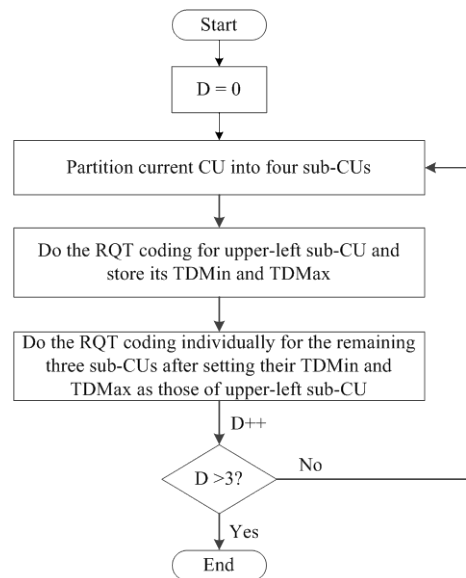


Fig. 3. Flowchart of the FCS-ET algorithm. D: CU depth level; TDMIN: minimum transform depth level; TDMAX: maximum transform depth level.

The flowchart of FCS-ET algorithm is depicted in **Fig. 3**. Firstly, for each treeblock (i.e., CU depth level equals to 0), perform CU quadtree partitioning at each CU depth level (i.e., 0~3) as the original HEVC encoder indicated. Secondly, perform separate operations for four sub-CUs generated from one CU quadtree partitioning as follows. For the first sub-CU (i.e., upper-left sub-CU), its minimum and maximum transform depth levels are derived and stored after completing the RQT coding. For the remaining three sub-CUs including the upper sub-CU and left sub-CU and right-bottom sub-CU, set their minimum and maximum transform depth levels as those of the first sub-CU prior to performing their respective RQT coding. Finally, implement intra/inter prediction and RQT coding for the remaining three sub-CUs. When the transform depth level of current sub-CU is smaller than the specified minimum transform depth level, the full check process is automatically skipped to perform the RQT coding at the next transform depth level. On the other hand, once the transform depth level of current sub-CU reaches the specified maximum transform depth level, the RQT coding is early terminated and no more RD cost evaluations will be conducted.

The FCS-ET algorithm is efficient for the computational complexity reduction because there exist a mass of CU quadtree partitionings in the encoding process of HEVC. The more CUs are quadtree partitioned, the more times the proposed FCS-ET algorithm will be performed, thus reducing more encoding time for HEVC encoders. The rationality of FCS-ET algorithm is verified using the same video sequences and test conditions mentioned in Section 3.1. The experimental results in **Table 3** show that more than 75% sub-CUs in each CU quadtree partitioning have a same transform depth level range. So it is reasonable to skip the full check process and early terminate the RQT coding for the last three sub-CUs in each CU quadtree partitioning. To keep consistent with the original HEVC encoder, the probability that the treeblocks, i.e., CU with size 64×64 , have a same transform depth level range is also shown in **Table 3**.

Table 3. The probability that sub-CUs in one CU quadtree partitionings have a same transform depth level range (%)

Sequences	N = 64	N = 32	N = 16	N = 8
BasketballDrive	76.4	89.1	82.9	69.2
Cactus	88.1	86.6	74.8	76.5
Johnny	80.4	79.2	88.5	77.6
KristenAndSara	79.8	68.3	81.1	71.1
BQMall	65.2	74.9	75.4	85.5
PartyScene	71.3	84.4	69.9	82.8
Average	76.9	80.4	78.8	77.1

3.3 Overall Algorithm

Finally, the adaptive maximum transform depth determination (AMTD) algorithm and full check skipping - early termination (FCS-ET) algorithm are joined together for fast TU size decision for the RQT coding in HEVC. The pseudo-code of the proposed overall algorithm is described in **Table 4**, where $GMaxTD$ represents the global maximum transform depth level for current treeblock. As can be seen from **Table 4**, the AMTD algorithm is applied on the global level, i.e., the treeblock level, while the FCS-ET algorithm is used for the local level, i.e., the CU level. Hence, the proposed overall algorithm is exploited to achieve large encoding time reduction on both the treeblock level and CU level.

Table 4. Pseudo-code of the overall algorithm for fast TU size decision

Overall Algorithm for fast TU size decision

```

1: // Encode each frame which contains  $M$  treeblocks
2: for  $1 \leq i \leq M$  do
3:   if there exist treeblocks to the left, to the upper and to the upper-left of treeblock  $i$  then
4:     // Perform AMTD and obtain  $GMaxTD$  for treeblock  $i$ 
5:   else
6:     // Perform the same operations as the original HEVC encoder
7:   end if
8:   // There are  $N$  CU quadtree partitionings for treeblock  $i$ 
9:   for  $1 \leq j \leq N$  do
10:    // Perform FCS-ET for  $j$  CU quadtree partitioning, subject to the condition that the
        maximum transform depth levels of all sub-CUs in the quadtree partitioning are
        smaller than  $GMaxTD$ 
11:     $j++$ 
12:  end for
13:   $i++$ 
14: end for

```

4. Experimental Results and Analysis

In order to evaluate the performance of the proposed TU size decision method for the RQT coding in HEVC, the AMTD algorithm, the FCS-ET algorithm, and the overall algorithm are individually implemented on the HEVC test model reference software, HM 13.0 [24]. Experimental results are shown in the following section. Various video sequences are tested under the common test conditions defined in [23]: Two coding configurations are tested for all video sequences, i.e., RA and LB, which corresponds to a broadcast scenario with a maximum GOP (Group of Picture) size of 8 and to a low-delay scenario with no picture reordering, respectively. The treeblock consists of 64×64 luma samples together with two corresponding blocks of chroma samples. CU size can vary from 64×64 down to 8×8 and TU supports transform block size from 4×4 to 32×32 . Five classes of video sequences with different resolutions are tested: class A (4K \times 2K), class B (1080p), class C (WVGA), class D (QWVGA) and class E (720p). To show how different bitrates impact on the TU size decision, each sequence is encoded with four QPs 22, 27, 32 and 37. The coding efficiency is evaluated with peak signal noise ratio (PSNR) and bitrate, and computational complexity is measured with the consumed encoding time. BDPSNR (dB) and BDBR (%) [25] are used to represent the averaged PSNR and bitrate differences, and positive and negative values represent increments and decrements, respectively. The time savings are calculated as

$$\Delta T = \frac{T_{proposed} - T_{anchor}}{T_{anchor}} \times 100\% \quad (3)$$

where $T_{proposed}$ denotes the overall encoding time of the proposed fast algorithm, and T_{anchor} denotes the total encoding time of the anchor HM 13.0.

Table 5 and **Table 6** show individual evaluation results of the AMTD algorithm and the FCS-ET algorithm under RA and LB configuration, respectively. As shown in **Table 5** and **Table 6**, the AMTD algorithm can achieve 5.6% and 4.9% encoding time reduction for all tested sequences under RA and LB configuration, respectively. And the coding efficiency loss

introduced by the AMTD algorithm is very negligible, i.e., 0.003dB-0.004dB PSNR drop and 0.08%-0.12% bitrate increase. Therefore the AMTD algorithm can skip some unnecessary transform depth levels on the treeblock level to reduce the computational complexity requested by the RQT coding.

As far as the FCS-ET algorithm is concerned, 17.2% and 14.0% encoding time has been reduced under RA and LB configuration, respectively, with a maximum of 18.9% in “BlowingBubbles” (416×240, RA) and a minimum of 11.6% in “Johnny” (1280×720, LB). In addition, the averaged PSNR drops are 0.031dB and 0.027dB, and the averaged bitrate increase are 0.87% and 0.78% for all tested sequences under RA and LB configurations, respectively. Few PSNR drop and bitrate increase indicate the coding efficiency loss by the FCS-ET algorithm is negligible. It is obvious that the FCS-ET algorithm can reduce more encoding time than the AMTD algorithm. The reason is that the FCS-ET algorithm is applied on the basis of CU, while the AMTD algorithm is based on treeblock. The number of CU is much more than that of treeblock as we can see in Fig. 1. Therefore, the FCS-ET algorithm is used more frequently to save more encoding time.

The method in [21] is implemented on the HM 13.0 for fair comparison and the experimental results are also presented in Table 5 and Table 6. It can be seen that with the method [21], 11.8%~14.7% encoding time can be reduced with 0.74~0.79 bitrate increase and 0.023~0.025 PSNR drop. Therefore, the FCS-ET algorithm outperforms the method [21] by achieving 2.5% more encoding time reduction without significant coding efficiency loss.

Table 5. Individual evaluation results of AMTD, FCS-ET and [21] under RA configuration

Sequences	Picture Size	AMTD			FCS-ET			[21]		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
PeopleOnStreet	2560x1600	0.10	-0.004	-6.06	0.81	-0.036	-17.71	0.78	-0.041	-15.69
Traffic	2560x1600	0.06	-0.002	-6.45	0.92	-0.030	-15.22	0.67	-0.020	-13.38
BasketballDrive	1920x1080	0.12	-0.003	-5.39	0.90	-0.020	-15.36	0.81	-0.018	-12.86
BQTerrace	1920x1080	0.11	-0.002	-5.98	1.03	-0.017	-17.32	0.90	-0.019	-14.78
Cactus	1920x1080	0.09	-0.002	-5.74	0.86	-0.019	-16.89	0.75	-0.016	-15.19
Kimonol	1920x1080	0.21	-0.006	-5.60	0.79	-0.029	-15.61	0.75	-0.011	-13.15
ParkScene	1920x1080	0.11	-0.004	-6.16	0.74	-0.023	-15.41	0.81	-0.026	-12.89
RaceHorses	832x480	0.02	-0.001	-5.68	0.83	-0.030	-17.34	0.66	-0.020	-15.46
BasketballDrill	832x480	0.06	-0.003	-5.81	0.88	-0.036	-18.17	0.54	-0.019	-16.14
BQMall	832x480	0.08	-0.003	-6.51	1.12	-0.043	-17.04	0.87	-0.041	-14.46
PartyScene	832x480	0.06	-0.002	-6.11	0.81	-0.035	-18.66	0.76	-0.029	-16.35
RaceHorses	416x240	0.03	-0.001	-4.42	0.82	-0.038	-18.47	0.68	-0.033	-13.62
BasketballPass	416x240	0.04	-0.002	-4.86	0.86	-0.040	-17.92	0.78	-0.030	-15.16
BlowingBubbles	416x240	0.07	-0.003	-4.94	0.86	-0.035	-18.94	0.69	-0.031	-15.79
BQSquare	416x240	0.03	-0.001	-4.47	0.84	-0.032	-17.98	0.70	-0.027	-15.55
Average		0.08	-0.003	-5.61	0.87	-0.031	-17.20	0.74	-0.025	-14.70

Table 6. Individual evaluation results of AMTD, FCS-ET and [21] under LB configuration

Sequences	Picture Size	AMTD			FCS-ET			[21]		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
BasketballDrive	1920x1080	0.14	-0.004	-5.17	0.67	-0.016	-13.96	0.62	-0.016	-10.82
BQTerrace	1920x1080	0.18	-0.003	-6.62	0.62	-0.011	-14.95	0.65	-0.018	-12.27
Cactus	1920x1080	0.13	-0.003	-5.78	0.61	-0.015	-14.40	0.57	-0.013	-11.94
Kimonol	1920x1080	0.33	-0.010	-4.32	0.64	-0.010	-12.83	0.60	-0.009	-10.42
ParkScene	1920x1080	0.09	-0.003	-5.23	0.67	-0.021	-13.57	0.74	-0.024	-11.33
FourPeople	1280x720	0.06	-0.004	-5.83	0.70	-0.024	-13.19	0.66	-0.020	-11.38

Johnny	1280x720	0.20	-0.001	-4.92	0.77	-0.017	-11.58	0.76	-0.017	-9.86
KristenAndSara	1280x720	0.08	-0.002	-4.86	0.53	-0.015	-11.95	0.50	-0.010	-9.79
RaceHorses	832x480	0.09	-0.003	-4.64	0.92	-0.035	-13.69	0.88	-0.032	-11.12
BasketballDrill	832x480	0.10	-0.004	-4.45	0.83	-0.033	-14.42	0.78	-0.033	-11.93
BQMall	832x480	0.13	-0.005	-5.19	0.86	-0.033	-13.90	0.84	-0.040	-12.29
PartyScene	832x480	0.09	-0.004	-5.19	1.01	-0.042	-16.19	0.99	-0.036	-13.26
RaceHorses	416x240	0.03	-0.001	-3.83	0.90	-0.042	-14.58	0.91	-0.041	-11.83
BasketballPass	416x240	0.05	-0.002	-3.24	0.91	-0.043	-14.54	0.80	-0.038	-12.24
BlowingBubbles	416x240	0.06	-0.002	-4.19	1.12	-0.044	-16.90	1.05	-0.047	-13.69
BQSquare	416x240	0.03	-0.001	-4.69	1.29	-0.045	-16.39	1.21	-0.046	-14.21
Average		0.12	-0.004	-4.90	0.78	-0.027	-14.04	0.79	-0.023	-11.77

Table 7. Results of the proposed overall algorithm compared with the RQT-AMS algorithm [20] under RA configuration

Sequences	Picture Size	Overall Algorithm			RQT-AMS		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
PeopleOnStreet	2560x1600	1.44	-0.062	-22.35	0.82	-0.042	-10.82
Traffic	2560x1600	1.24	-0.040	-20.85	0.67	-0.034	-12.10
BasketballDrive	1920x1080	1.16	-0.025	-19.41	0.80	-0.047	-10.55
BQTerrace	1920x1080	1.33	-0.022	-23.03	0.63	-0.032	-9.87
Cactus	1920x1080	1.11	-0.025	-21.25	0.75	-0.035	-11.12
Kimonol	1920x1080	0.52	-0.015	-19.62	0.68	-0.044	-10.47
ParkScene	1920x1080	1.07	-0.033	-20.63	0.79	-0.039	-13.20
RaceHorses	832x480	1.33	-0.048	-22.03	0.82	-0.041	-10.91
BasketballDrill	832x480	1.07	-0.043	-21.91	0.66	-0.033	-10.32
BQMall	832x480	1.50	-0.057	-21.47	0.76	-0.037	-12.83
PartyScene	832x480	1.06	-0.045	-23.34	0.61	-0.031	-12.35
RaceHorses	416x240	1.24	-0.056	-22.04	0.74	-0.037	-10.09
BasketballPass	416x240	1.12	-0.052	-21.65	0.72	-0.034	-11.89
BlowingBubbles	416x240	1.18	-0.048	-23.38	0.82	-0.040	-9.27
BQSquare	416x240	1.33	-0.049	-22.38	0.69	-0.035	-10.80
Average		1.18	-0.041	-21.69	0.73	-0.037	-11.12

Table 8. Results of the proposed overall algorithm compared with the RQT-AMS algorithm [20] under LB configuration

Sequences	Picture Size	Overall Algorithm			RQT-AMS		
		BDBR (%)	BDPSNR (dB)	ΔT (%)	BDBR (%)	BDPSNR (dB)	ΔT (%)
BasketballDrive	1920x1080	1.41	-0.032	-22.20	0.87	-0.045	-18.83
BQTerrace	1920x1080	1.29	-0.029	-21.95	0.67	-0.034	-17.79
Cactus	1920x1080	1.16	-0.025	-20.62	0.78	-0.039	-18.03
Kimonol	1920x1080	0.96	-0.032	-21.25	0.83	-0.040	-16.85
ParkScene	1920x1080	1.51	-0.048	-20.24	0.79	-0.039	-17.58
FourPeople	1280x720	1.28	-0.043	-19.44	0.63	-0.032	-16.21
Johnny	1280x720	1.26	-0.028	-20.87	0.62	-0.030	-16.53
KristenAndSara	1280x720	0.94	-0.025	-21.43	0.88	-0.053	-18.37
RaceHorses	832x480	1.21	-0.045	-20.10	0.79	-0.040	-17.99
BasketballDrill	832x480	1.52	-0.059	-19.04	0.81	-0.041	-17.17
BQMall	832x480	1.79	-0.071	-21.08	0.82	-0.039	-18.20
PartyScene	832x480	1.07	-0.048	-21.47	0.77	-0.039	-16.72
RaceHorses	416x240	1.12	-0.050	-20.20	0.63	-0.032	-16.77
BasketballPass	416x240	0.95	-0.047	-20.66	0.65	-0.033	-19.01
BlowingBubbles	416x240	1.19	-0.046	-21.01	0.63	-0.030	-17.44
BQSquare	416x240	1.50	-0.049	-20.22	0.76	-0.038	-17.84
Average		1.24	-0.042	-20.77	0.75	-0.038	-17.57

The experimental results of the comparison of the proposed overall algorithm presented in Section 3.3 and a fast RQT algorithm for HEVC encoders (adaptive RQT mode selection algorithm, RQT-AMS [20]) under RA and LB configuration are shown in Table 7 and Table 8, respectively, where both two algorithms are implemented on HM 13.0. From Table 7 and Table 8 we can see that, the proposed overall algorithm can reduce on average 21.7% and 20.8% encoding time under RA and LB configuration, respectively, and outperform the RQT-AMS algorithm in terms of the computational complexity with almost the same coding efficiency. The reason is that compared to the RQT-AMS algorithm which is based on CU level to speed up the RQT coding, the proposed overall algorithm can be applied on both treeblock level and CU level so as to reduce a large number of encoding time. Also shown is a consistent gain in encoding time for all tested sequences with a lowest gain of 19.4% for “BasketballDrill” (832×480, LB) and a highest gain of 23.4% for “BlowingBubbles” (416×240, RA). Besides, the coding efficiency loss introduced by the proposed overall algorithm is negligible, the averaged PSNR loss is 0.041-0.042 dB and the averaged bitrate increase is 1.18-1.24% for all tested sequences.

In Fig. 4(a), rate-distortion curves of the algorithms including: the AMTD algorithm, the FCS-ET algorithm, the proposed overall algorithm and the original RQT coding, are depicted for the “PeopleOnStreet” (2560×1600), in which the PSNR is plotted as a function of the average bitrate. All algorithms are implemented on HM 13.0. As shown in Fig. 4(a), all the rate-distortion curves are clustered together, which means the proposed algorithms, i.e., the AMTD algorithm, the FCS-ET algorithm and the proposed overall algorithm are similar to the original RQT coding in HM 13.0 encoder in terms of the coding efficiency. The bitrate savings of the proposed algorithms relative to the original RQT coding in HM 13.0 encoder under different QPs are plotted in Fig. 4(b). Conclusions can be drawn from Fig. 4(b) that the proposed algorithms can achieve a consistent encoding time saving over a large bitrate range with almost negligible loss in coding efficiency compared with the original RQT coding in HM 13.0 encoder.

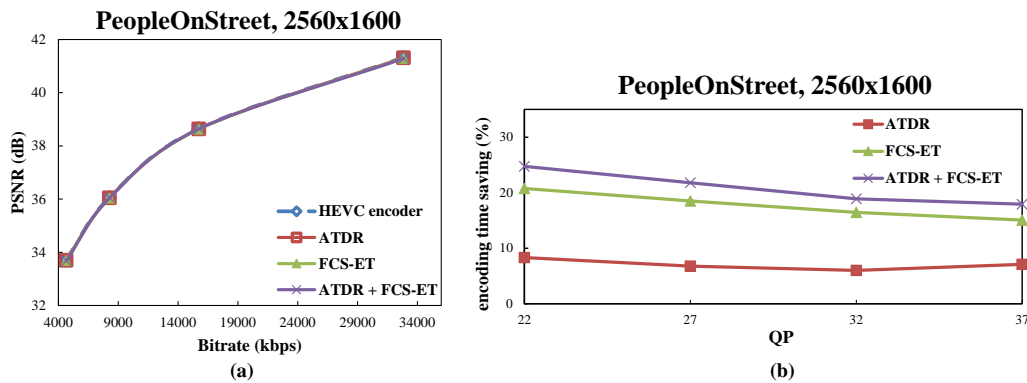


Fig. 4. Experimental results of “PeopleOnStreet” (2560 × 1600) under different QPs (22, 27, 32, 37). (a) RD curves of “PeopleOnStreet”. (b) Time saving curves of “PeopleOnStreet”.

In addition, the subjective quality comparisons are conducted to further validate the effectiveness of the proposed overall algorithm. Each test sequence is encoded at QP = 22, 27, 32, and 37 using both HM 13.0 and the proposed overall algorithm. Fig. 5 and Fig. 6 shows the subjective quality comparisons of the 126th frame of “PeopleOnStreet” (2560 × 1600) and the 204th frame of “PartyScene” (832 × 480), respectively. Specifically, “PeopleOnStreet” is

encoded under RA configuration, while “PartyScene” is encoded under LB configuration. The averaged PSNR is calculated as the weighted sum of the PSNR per picture of the individual components ($PSNR_Y$, $PSNR_U$, and $PSNR_V$) [3]

$$PSNR = (6 \cdot PSNR_Y + PSNR_U + PSNR_V) / 8 \quad (4)$$

As shown in Fig. 5 and Fig. 6, compared to HM 13.0, the subjective quality of the sequences (e.g., “PeopleOnStreet” and “PartyScene”) encoded using the proposed overall algorithm degrades little under different QPs (the largest PSNR loss is 0.2 dB). From Fig. 5 and Fig. 6 we can also conclude that, the proposed overall algorithm is effective for different video resolutions and coding configurations in terms of subjective quality.



(a) QP = 22. PSNR(left): 41.62 dB; PSNR(right): 41.51 dB; PSNR loss: 0.11 dB.



(b) QP = 27. PSNR(left): 38.88 dB; PSNR(right): 38.79 dB; PSNR loss: 0.09 dB.



(c) QP = 32. PSNR(left): 36.25 dB; PSNR(right): 36.15 dB; PSNR loss: 0.10 dB.



(d) QP = 37. PSNR(left): 33.86 dB; PSNR(right): 33.76 dB; PSNR loss: 0.10 dB.

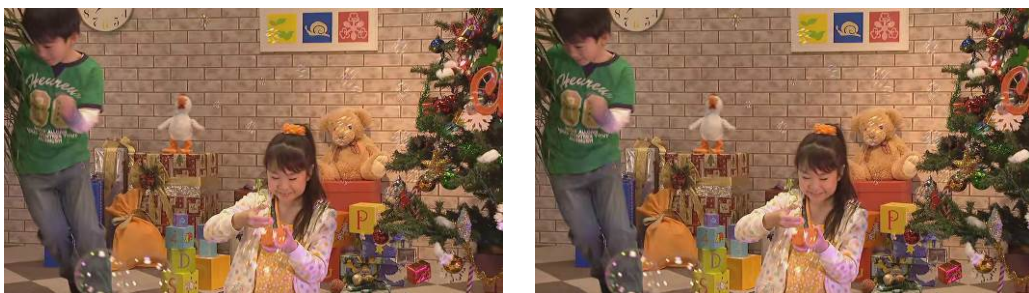
Fig. 5. Subjective quality comparisons of the 126th frame of “PeopleOnStreet” (2560×1600) under different QPs in RA case. The encoded pictures using HM 13.0 and the proposed overall algorithm are presented in the left and right columns, respectively.



(a) QP = 22. PSNR(left): 40.31 dB; PSNR(right): 40.21 dB; PSNR loss: 0.10 dB.



(b) QP = 27. PSNR(left): 36.53 dB; PSNR(right): 36.38 dB; PSNR loss: 0.15 dB.



(c) QP = 32. PSNR(left): 33.14 dB; PSNR(right): 32.95 dB; PSNR loss: 0.19 dB.



(d) QP = 37. PSNR(left): 30.08 dB; PSNR(right): 29.91 dB; PSNR loss: 0.17 dB.

Fig. 6. Subjective quality comparisons of the 204th frame of “PartyScene” (832×480) under different QPs in LB case. The encoded pictures using HM 13.0 and the proposed overall algorithm are presented in the left and right columns, respectively.

5. Conclusion

In this paper, we propose an effective TU size decision method for the RQT coding in HEVC, which incorporates an adaptive maximum transform depth determination (AMTD) algorithm and a full check skipping - early termination (FCS-ET) algorithm, to reduce the computational complexity requested by the RQT coding. The recent HEVC test model reference software, HM 13.0, is applied to evaluate the proposed overall algorithm. The experimental results show that the proposed overall algorithm can significantly reduce the computational complexity requested by the RQT coding while maintaining almost the same RD performance as HM 13.0 encoder. Meanwhile, the proposed overall algorithm outperforms a state-of-the-art fast TU size decision method, i.e., RQT-AMS algorithm. The experimental results indicate that our proposed overall algorithm should be considered in the design of a fast HEVC encoder.

For future work, we plan to apply the proposed algorithm to a practical video encoder. A practical video encoder needs low computational complexity, acceptable visual quality, and good channel assignment, rate allocation strategy. Note that there have been some recent developments for mobile network video coding [26, 27], which can be useful for designing a practical video coder.

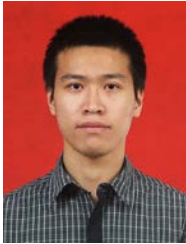
References

- [1] G. J. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, 2012. [Article \(CrossRef Link\)](#)
- [2] J. Ohm, G. J. Sullivan, “High efficiency video coding: The next frontier in video compression [standards in a nutshell],” *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 152-158, 2013. [Article \(CrossRef Link\)](#)
- [3] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, T. Wiegand, “Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC),” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669-1684, 2012. [Article \(CrossRef Link\)](#)
- [4] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alshin, T. Lee, *et al.*, “Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1709-1720, 2010. [Article \(CrossRef Link\)](#)
- [5] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, *et al.*, “Transform Coding

- Techniques in HEVC,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 978-989, 2013. [Article \(CrossRef Link\)](#)
- [6] F. Bossen, B. Bross, K. Suhring, D. Flynn, “HEVC complexity and implementation analysis,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685-1696, 2012. [Article \(CrossRef Link\)](#)
- [7] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, M. Sadafale, “Core Transform Design for the High Efficiency Video Coding (HEVC) Standard,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1029-1041, 2013. [Article \(CrossRef Link\)](#)
- [8] “High Efficiency Video Coding (HEVC) Test Model 13 (HM13) Encoder Description,” *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, Document: JCTVC-O1002, 15th Meeting, 2013.
- [9] K. Choi, E. S. Jang, “Fast coding unit decision method based on coding tree pruning for high efficiency video coding,” *Optical Engineering*, vol. 51, pp. 030502-1-030502-3, 2012. [Article \(CrossRef Link\)](#)
- [10] X. Shen, L. Yu, J. Chen, “Fast coding unit size selection for HEVC based on Bayesian decision rule,” in *Proc. of Picture Coding Symposium (PCS)*, pp. 453-456, 2012. [Article \(CrossRef Link\)](#)
- [11] W. Jiang, H. Ma, Y. Chen, “Gradient based fast mode decision algorithm for intra prediction in HEVC,” in *Proc. of Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, pp. 1836-1840, 2012. [Article \(CrossRef Link\)](#)
- [12] L. Shen, Z. Liu, X. Zhang, W. Zhao, Z. Zhang, “An effective CU size decision method for HEVC encoders,” *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 465-470, 2013. [Article \(CrossRef Link\)](#)
- [13] L. Shen, Z. Zhang, P. An, “Fast CU size decision and mode decision algorithm for HEVC intra coding,” *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 207-213, 2013. [Article \(CrossRef Link\)](#)
- [14] L. Zhao, L. Zhang, S. Ma, D. Zhao, “Fast mode decision algorithm for intra prediction in HEVC,” in *Visual Communications and Image Processing (VCIP), 2011 IEEE*, pp. 1-4, 2011. [Article \(CrossRef Link\)](#)
- [15] H. Zhang, Z. Ma, “Fast intra prediction for high efficiency video coding,” in *Advances in Multimedia Information Processing-PCM 2012*, ed: Springer, pp. 568-577, 2012.
- [16] H. Zhang and Z. Ma, “Fast Intra Mode Decision for High-Efficiency Video Coding (HEVC),” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 660-668, Apr. 2014. [Article \(CrossRef Link\)](#)
- [17] Y. H. Tan, C. Yeo, H. L. Tan, Z. Li, “On residual quad-tree coding in HEVC,” in *Proc. of Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, pp. 1-4, 2011. [Article \(CrossRef Link\)](#)
- [18] S.-W. Teng, H.-M. Hang, Y.-F. Chen, “Fast mode decision algorithm for residual quadtree coding in HEVC,” in *Visual Communications and Image Processing (VCIP), 2011 IEEE*, pp. 1-4, 2011. [Article \(CrossRef Link\)](#)
- [19] K. Choi, E. S. Jang, “Early TU decision method for fast video encoding in high efficiency video coding,” *Electronics letters*, vol. 48, pp. 689-691, 2012. [Article \(CrossRef Link\)](#)
- [20] Y. Zhang, M. Zhao, “An adaptive RQT mode selection algorithm for HEVC,” in *Proc. of Image and Signal Processing (CISP), 2012 5th International Congress on*, pp. 173-177, 2012. [Article \(CrossRef Link\)](#)
- [21] J. F. Wu, B. L. Guo, J. Jiang, “Fast TU Size Decision Algorithm for HEVC,” *ICIC Express Letters*, vol. 8, pp. 2327-2333, 2014.
- [22] B. Lee, M. Kim, “Modeling rates and distortions based on a mixture of Laplacian distributions for inter-predicted residues in quadtree coding of HEVC,” *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 571-574, 2011. [Article \(CrossRef Link\)](#)
- [23] F. Bossen, “Common test conditions and software reference configurations,” *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, Document: JCTVC-L1100, 12th Meeting: Geneva, CH, 2013.
- [24] JCT-VC (2014) Subversion repository for the HEVC test model reference software.

https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.

- [25] G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," VCEG-M33, Austin, TX, USA, 2001.
- [26] L. Zhou, X. B. Wang, W. T. G. Muntean, B. Geller, "Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 409-419, 2010. [Article \(CrossRef Link\)](#)
- [27] L. Zhou, Q. Y. Hu, Y. Qian, H. H. Chen, "Energy-Spectrum Efficiency Tradeoff for Video Streaming over Mobile Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 5, pp. 981-991, 2013. [Article \(CrossRef Link\)](#)



Jinfu Wu received his B.S. degrees in control technology and instrument from Xidian University, Xi'an, P.R. China in 2011. He is currently a Ph.D. student with the School of Aerospace Science and Technology, Xidian University. His research interests include video coding and image processing.



Baolong Guo received his B.S., M.S. and Ph.D. degrees from Xidian University in 1984, 1988 and 1995, respectively, all in communication and electronic system. From 1998 to 1999, he was a visiting scientist at Doshisha University, Japan. He is currently the director of the Institute of Intelligent Control & Image Engineering (ICIE) at Xidian University. His research interests include neural networks, pattern recognition, intelligent information processing, and image communication.



Yunyi Yan received his B.S. degree in automation, M.S. degree in pattern recognition and intelligent system, and Ph.D degree in electronic science and technology from Xidian University in 2001, 2004 and 2008 respectively in China. Since 2004, he has been with ICIE Institute of Aerospace Science and Technology School of Xidian University, where he is currently an associated professor. Since 2013, he served as the director of Intelligent Detection Department in Xidian University. He worked as a visiting scholar in Montreal Neurological Institute of McGill University, Canada from 2010 to 2011. His research interests include system reliability analysis, medical image processing, and swarm intelligence.



Jie Hou obtained B.S. degree in Automation from Xidian University, P.R.China in 2011. Currently, He's pursuing Ph.D. degree in Measuring Technology and Instruments at Xidian University. His research mainly focuses on video processing and object recognition.



Dan Zhao received his B.S. degrees in control technology and instrument from Xidian University, Xi'an, P.R. China in 2012. He is currently a Ph.D. student with the school of Aerospace Science and Technology, Xidian University. His research interests include video processing and object tracking based on online learning.