Journal of Electronic Imaging

JElectronicImaging.org

Computation reduction in high-efficiency video coding based on the similarity of transform unit blocks

Zong-Yi Chen
Jiunn-Tsair Fang
Chung-Shian Chiang
Pao-Chi Chang

Computation reduction in high-efficiency video coding based on the similarity of transform unit blocks

Zong-Yi Chen,[a] Jiunn-Tsair Fang,[b] Chung-Shian Chiang,[a] and Pao-Chi Chang[a,*]
[a]National Central University, Department of Communication Engineering, No. 300, Jhongda Road, Jhongli City, 32001 Taiwan
[b]Ming Chuan University, Department of Electronic Engineering, No. 5, Deming Road, Taoyuan County, 33348 Taiwan

Abstract. The new video coding standard, high-efficiency video coding, adopts a quadtree structure to provide variable transform sizes in the transform coding process. The heuristic examination of transform unit (TU) modes substantially increases the computational complexity, compared to previous video coding standards. Thus, efficiently reducing the TU candidate modes is crucial. In the proposed similarity-check scheme, sub-TU blocks are categorized into a strongly similar case or a weakly similar case, and the early TU termination or early TU splitting procedure is performed. For the strongly similar case, a property called zero-block inheritance combined with a zero-block detection technique is applied to terminate the TU search process early. For the weakly similar case, the gradients of residuals representing the similarity of coefficients are used to skip the current TU mode or stop the TU splitting process. In particular, the computation time is further reduced because all the required information for the proposed mode decision criteria is derived before performing the transform coding. The experimental results revealed that the proposed algorithm can save ~64% of the TU encoding time on average in the interprediction, with a negligible rate-distortion loss. *© 2014 SPIE and IS&T* [DOI: 10.1117/1.JEI.23.6.061105]

Keywords: complexity reduction; fast algorithm; high-efficiency video coding; mode decision; transform unit.

Paper 14173SS received Apr. 28, 2014; revised manuscript received Jul. 9, 2014; accepted for publication Jul. 17, 2014; published online Aug. 20, 2014.

1 Introduction

Video coding technologies have been developed for several decades, and many video compression standards have been established. The source data in video encoding systems are typically compressed into small representations for storage or transmission. With the increasing demand for high-quality and high-resolution video content, the Joint Collaborative Team on Video Coding (JCT-VC) started to establish a new video coding standard called high-efficiency video coding (HEVC)[1] in 2010. HEVC was aimed at saving half of the bit rate of H.264/AVC[2] while maintaining the same subjective video quality and was officially finalized as an international standard in April 2013. The objective of these standards is to compress the source data as small bitrates, without sacrificing too much video quality. Thus, recent video coding technologies are often accompanied by complicated prediction processes, which result in considerable encoding complexity. However, for real-time video encoding systems or power-constrained mobile devices, complexity reduction is required. Therefore, increasingly fast algorithms have been proposed to decrease the encoding complexity of recent standards.

Similar to previous video coding standards, most of the encoding complexity of a video encoder is consumed by the interprediction. Many fast algorithms[3–14] have been proposed for the newest HEVC video encoder. In Refs. 3 and 4, the authors used correlations between current and spatiotemporal coding units (CUs) to skip the rarely used CU depths. Based on the temporal similarity between the current CU and its colocated block, a maximum of two prediction unit (PU) modes were selected for each CU depth in Ref. 5. The spatial, temporal, and depth correlations of neighboring CUs were used to calculate the block motion complexity of the current CU to reduce the PU candidates for each CU depth in Ref. 6. Furthermore, a complexity reduction algorithm for an HEVC fractional-pixel motion estimation was proposed in Ref. 7. Wang et al.[8] proposed a fast multireference frame motion estimation algorithm. Other studies[9–11] have concentrated on accelerating the intraprediction. In addition to fast encoders, researchers have developed complexity control mechanisms[12–14] that reduced the encoding time and satisfied the complexity requirements constrained by devices or applications. However, most of the proposed algorithms focused on the efficient CU/PU mode decision or motion estimation strategies.

In HEVC, the basic encoding unit that plays a similar role to the macroblock in H.264/AVC is separated into a CU, PU, and transform unit (TU). The TU executes the transform, quantization, and entropy coding processes in HEVC. Unlike the use of the fixed-size transform in previous video coding standards, a residual quadtree (RQT), which provides the nested quadtree-based transform coding, is adopted in the TU of HEVC. The quadtree structure uses variable transform block sizes from 4×4 to 32×32 to adapt to the various characteristics of the prediction blocks. However, the encoder must test all possible TU candidate modes (i.e., the possible transform sizes) to determine the optimal method to achieve efficient compression. The heuristic examination of TU modes leads to considerable computational complexity in the encoder. The encoding complexity of the TU[15] represents ~30% of the total encoding time, and it cannot be ignored when developing fast

*Address all correspondence to: Pao-Chi Chang, E-mail: pcchang@ce.ncu.edu.tw

encoding algorithms. An efficient TU mode decision algorithm is, thus, crucial for HEVC.

Some studies have concentrated on fast mode algorithms related to the transform part. The literature[16,17] has provided zero-block detection methods by theoretically analyzing the lower limit and upper limit of thresholds for a zero-block in H.264/AVC. A zero-block implies that all coefficients in a prediction block are zeroes after quantization. Once a zero-block is detected, the remaining transform and quantization processes of this zero-block can be skipped. Unlike fast encoding algorithms for CU/PU, studies on fast TU mode decision for HEVC are limited. Teng et al.[18] proposed a fast RQT mode decision algorithm for HEVC by considering the rate-distortion (RD) efficiency. A property called zero-block inheritance (ZBI) was observed in Ref. 18 and indicates that if a TU is a zero-block, its four sub-TUs are likely to all be zero-blocks. In Ref. 19, the number of nonzero coefficients of the root TU was used to terminate the subtree RQT processes in HEVC. In Ref. 20, the position of the last nonzero transform coefficient and the number of zero transform coefficients were used to decide whether a TU should be split or not. Zhang et al.[21] set the maximum inter-RQT depth to 1 when the CU size was 64×64 and set the maximum inter-RQT depth to 2 when the CU size was 8×8. For other CU sizes, a classifier based on discriminant analysis was used to determine whether the maximum inter-RQT depth was 1 or 3. In Ref. 22, an early TU split termination scheme was proposed based on the concept of quasi-zero-block (QZB). QZB was determined by the sum of all absolute quantized transform coefficients and the number of nonzero coefficients. The maximum depth of RQT was set to 2 according to the analysis of TU depth statistics.

The previously proposed algorithms[18–20,22] for HEVC simplified the RQT processes by applying RD costs or the coefficients. However, these algorithms used the information after discrete cosine transform (DCT) and quantization for achieving their fast decisions. The encoding of at least one TU mode is essential. Another method proposed in Ref. 21 might require online training and updating processes to improve the hit rate of classification. Instead of using the encoded information in the frequency domain, all the required information for the proposed mode decision criteria in this work was derived before performing transform coding to further decrease the computational complexity. In this paper, a similarity-check scheme was proposed for the inter-prediction of TU mode (inter-TU). The residual of sub-blocks were categorized into the strongly similar case and weakly similar case, and the early TU termination or early TU splitting procedure was performed. For the strongly similar case, the threshold for a zero-block was derived based on the magnitude of the DC coefficient. Then the zero-block detection technique was further combined with the ZBI to terminate the TU search process early. For the weakly similar case, the gradients of residuals representing the similarity of coefficients were used to skip the current TU mode or stop the TU splitting process.

This paper is organized as follows. Section 2 introduces the transform, quantization, and RQT in HEVC. Section 3 presents the proposed algorithm in detail. The experimental results are described in Sec. 4. Finally, Sec. 5 provides a conclusion.

Table 1 The relation between coding unit (CU) depth and transform unit (TU) size.

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

2 Transform Unit in High-Efficiency Video Coding

The TU decision process consists of transform, quantization, and entropy coding in HEVC. TU has variable block sizes, which are dependent on the CU depth (Table 1). The default maximum TU depth is 3, and its corresponding maximum size is 32×32. The RQT structure of the TU provides the nested quadtree-based transform coding for various characteristics of prediction residuals. In this section, the transform, quantization, and RQT in the HM (the HEVC test model) are briefly introduced.

2.1 *Transform and Quantization*

The transform of the TU is described as follows. Given an $N \times N$ residual block X, the transform coefficient F is calculated from the first dimension operation XC^T and the second dimension operation $C(XC^T)$ as follows:

$$F = CXC^{\mathrm{T}}, \tag{1}$$

where

$$X = \begin{bmatrix} x(0,0) & \cdots & x(0,N-1) \\ \vdots & x(i,j) & \vdots \\ x(N-1,0) & \cdots & x(N-1,N-1) \end{bmatrix},$$

$$C = \begin{bmatrix} c(0,0) & \cdots & c(0,N-1) \\ \vdots & c(i,j) & \vdots \\ c(N-1,0) & \cdots & c(N-1,N-1) \end{bmatrix}$$

is the transform matrix, T denotes the transpose of the matrix, and N is the TU size. The outputs of the transform operation are $T_{2D}(i,j)$.

The quantized transform coefficients $\mathrm{Coeff}(i,j)$ are derived from the transform coefficients $T_{2D}(i,j)$, which can be expressed by a function of $T_{2D}(i,j)$, quantization parameter (QP), and N as follows:

$$\mathrm{Coeff}(i,j) = q[T_{2D}(i,j), QP, N]. \quad (2)$$

2.2 *Residual Quadtree*

The transform operation converts residuals from the pixel domain into the frequency domain to compact energy and decorrelate the similarity among pixels. In general, a large transform block size is suitable for a still region because the energy is mostly located in the low-frequency parts. By contrast, a region with a complex texture and fast motion should be partitioned into small blocks to reduce the distortion. The RQT structure in HEVC provides a flexible combination of transform block sizes to achieve a more effective RD performance than that of a fixed-size transform. The prediction of each PU mode is finished and the corresponding residuals are ready before the TU is processed. For each candidate residual block, the current TU is partitioned into four sub-TUs until the partition reaches the allowable minimum TU size, as listed in Table 1. The rate-distortion optimization (RDO) of the TU modes is recursively done to compare the RD cost of the current TU and the sum of the RD costs of its four sub-TUs. An example of the full search of an RQT procedure is illustrated in Fig. 1(a), where the search orders are labeled in alphabetical order, and the optimal TU mode is determined by comparing the RD costs of all possible partition combinations. A TU partition example is presented in Fig. 1(b), where the sum of the RD costs of blocks h, i, j, and k is smaller than the RD cost of g, and block g is, thus, split.

To examine the complexity of the inter-TU in an HEVC encoder, a series of sequences were tested with four distinct QPs (22, 27, 32, and 37); 100 frames were encoded using HM14.0.[23] In this paper, the complexity was measured in terms of the running time of the HM encoder. For a TU block, the best TU mode is the one with the least RD cost by comparison among different TU modes. The corresponding running time is the accumulated time to perform each TU mode by the processes of transform, quantization, and RD cost calculation. HM was executed by Microsoft Visual Studio 2013 with Win32 platform and the release mode. From the results in Table 2, TU represented ~27% of the total encoding time, which was much higher than the time consumed by transform and quantization of previous standards. Such an exhaustive search is time consuming, and it is, thus, critical to design an efficient TU search algorithm to make real-time applications feasible.

Table 2 The proportion of TU encoding time. The configuration is lowdelay_P_main.

Sequence		TU time (%)
ClassB	BasketballDrive	23.32
	BQTerrace	28.15
ClassC	BasketballDrill	27.80
	RaceHorses	26.16
ClassD	BlowingBubbles	30.69
	BQSquare	33.07
ClassE	Vidyo1	24.52
	Vidyo4	23.51
Average		27.15

3 Similarity of the Transform Unit Blocks

In the proposed algorithm, the early TU termination or early TU splitting is applied to the current TU block based on the similarity among its four sub-TU blocks. The similarity between two blocks is defined as the difference of coefficients in these two blocks. Two blocks are said to be similar if the difference is smaller than a predefined threshold. An early TU termination means that the current TU is stopped from splitting into four sub-TUs and only the current TU in the RDO process is tested. An early TU splitting means the current TU is directly split into four sub-TUs and the RD cost calculation of the current TU is skipped. Figure 2(a)

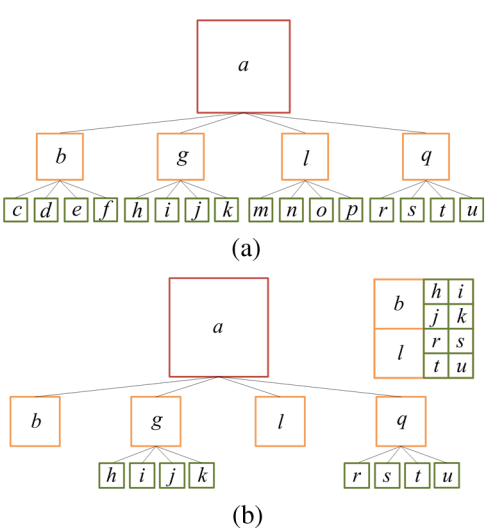

Fig. 1 Residual quadtree (RQT) illustration: (a) full RQT procedure. The search order is labeled in alphabetical order. (b) An example of RQT mode partition.

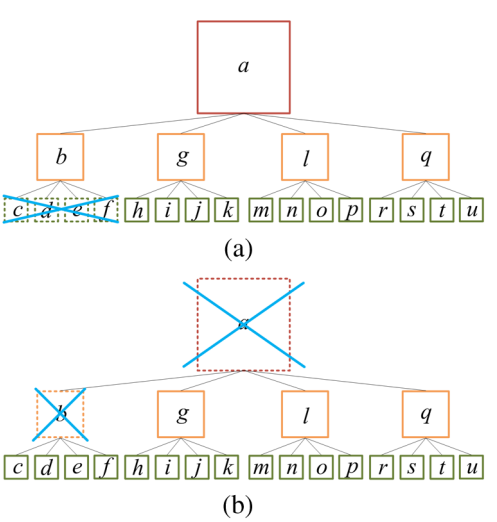

Fig. 2 Illustrations of two operations in the proposed fast transform unit (TU) mode decision: (a) early TU termination and (b) early TU splitting.

illustrates an example of an early TU termination, and Fig. 2(b) illustrates an early TU splitting.

The cost functions J for the current TU and its sub-TUs can be described as follows:

$$J_{\text{current}} = D_{\text{current}} + \lambda R_{\text{current}} = D_{\text{current}} + \lambda(R_{\text{residual,current}} + R_{\text{CBF,current}}), \quad (3)$$

$$J_{\text{sub}} = \sum_{i=1}^{4} D_{sub,i} + \lambda \sum_{i=1}^{4} R_{\text{sub},i} = \sum_{i=1}^{4} D_{\text{sub},i} + \lambda \sum_{i=1}^{4} (R_{\text{residual,sub},i} + R_{\text{CBF,sub},i}), \quad (4)$$

where D is the distortion, R is the rate for coding this TU, and λ is the Lagrange multiplier. The rate in the RD cost function consists of the bits for residual coding and coded block flag (CBF). In this study, when the coefficients of the sub-TUs are similar, i.e., the four sub-TUs are homogeneous, their distortions in the RD cost function are close, and the bits for residual coding are also close. In this case, the following is obtained:

$$D_{\text{current}} + \lambda R_{\text{residual,current}} \approx \sum_{i=1}^{4} D_{\text{sub},i} + \lambda \sum_{i=1}^{4} R_{\text{residual,sub},i}. \quad (5)$$

However, the CBF bits of the current TU are generally lower than the sum of the four sub-TUs.

$$R_{\text{CBF,current}} \leq \sum_{i=1}^{4} R_{\text{CBF,sub},i}. \quad (6)$$

If four sub-TUs are nearly the same, the rate term of the current TU in the RD cost function is lower than the rate term of the sub-TUs (i.e., $J_{\text{current}} \leq J_{\text{sub}}$). This indicates that selecting the current TU rather than the sub-TUs as the optimal mode is more likely to happen. By contrast, if the transform coefficients of these four sub-TUs are dissimilar, the RD performance may deteriorate if the current TU mode is applied. This is because the distortion may increase by adopting the current transform bases to represent these four distinct (dissimilar) coefficients in sub-TU blocks. Therefore, if the sub-TUs are not similar, the current TU is likely to split into four sub-TUs.

In this study, to further save the complexity, the similarity was calculated in the pixel domain. The degree of similarity was divided into strongly similar and weakly similar. The algorithms used to determine the similarity of these two cases are described in Secs. 3.1 and 3.2, respectively.

3.1 *Zero-Block Detection Combined with Zero Block Inheritance for the Strongly Similar Case*

Based on the discussion in Sec. 3, an extreme case can be first considered. When the current TU and its sub-TUs are all zero-blocks, the only different terms in the RD cost function are the CBF bits. In this case, the sub-TUs are strongly similar. To detect a zero-block before performing the transform coding, the threshold of a zero-block must be derived.

The concept of zero-block detection[17] used in H.264/AVC was applied to HEVC in this paper. Typically, the value of the DC coefficient is the largest among all the DCT coefficients. If the DC coefficient of a block is quantized to be zero, this block is likely a zero-block. In this study, a block was regarded as a zero-block if its quantized DC coefficient was zero.

Consequently, when the absolute value of the quantized value is <1, this coefficient is quantized into zero. Deriving the upper limit of a zero-block starts from $|\text{DC}| < 1$. By applying $|\text{Coeff}(0,0)| < 1$ in Eq. (2), the following can be obtained.

$$|q[T_{2D}(0,0), QP, N]| < 1. \quad (7)$$

The DC coefficient $T_{2D}(0,0)$ can be rearranged as the function of SAR and N as follows:

$$|T_{2D}(0,0)| = f(\text{SAR}, N), \quad (8)$$

where $\text{SAR} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i,j)|$ is the sum of the absolute residual intensity. Substituting Eq. (8) into Eq. (7), the upper limit of a zero-block can be derived as follows:

$$\text{SAR} < \text{TH}_{\text{ZB}}, \quad (9)$$

where TH_{ZB} is the threshold of a zero-block, which can be denoted as the function of QP and N.

$$\text{TH}_{\text{ZB}} = t(QP, N). \quad (10)$$

If the SAR of a block is smaller than TH_{ZB}, this block is assumed to be a zero-block. All the parameters used to calculate TH_{ZB} and SAR are those set before the transform coding.

ZBI (Ref. 18) is further combined with the zero-block detection in this paper. ZBI implies that if the current TU is a zero-block, its sub-TUs are likely to be all zero-blocks. Thus, once a block is detected to be a zero-block, its four sub-blocks are assumed to be zero-blocks. In this condition, the four sub-TUs are strongly similar because all sub-TUs are regarded as zero-blocks. The results of zero-block detection and ZBI can be applied to the proposed strongly similar case, in which the search of the remaining TU modes is terminated once a zero-block is detected.

To evaluate the accuracy of the proposed method, two sets are defined for a TU block: set $A = \{\text{zero-blocks}\}$ and set $B = \{\text{detected zero-blocks}\}$, and the corresponding complementary sets are denoted as A^C and B^C, respectively. Figure 3 illustrates the possible detection cases. The hit rate and failure rate are defined in Eqs. (11) and (12), respectively. Table 3 presents the statistics of the proposed zero-block detection. The result of each sequence is the average over four QPs (22, 27, 32, and 37).

$$\text{Hitrate} = \frac{\text{P}(A \cap B)}{\text{P}(B)}, \quad (11)$$

$$\text{Failure rate} = \frac{\text{P}(A \cap B^C)}{\text{P}(A)}. \quad (12)$$

The average hit rate of the tested sequences is >80%. It is observed that even if a block is error-detected to be a

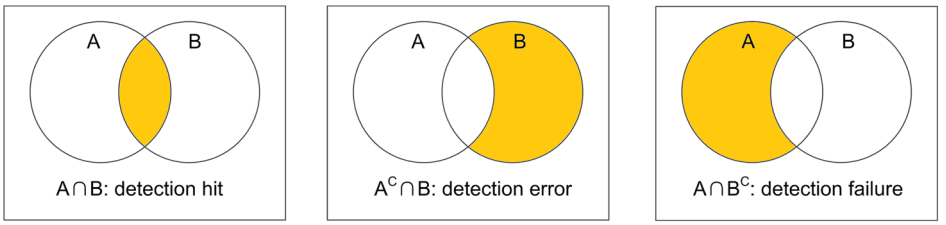

Fig. 3 Illustration of the zero-block detection cases.

Table 3 The statistics for evaluating the accuracy of zero-block detection. The configuration is lowdelay_P_main.

Sequence		Hit rate (%)	Failure rate (%)
ClassC	BasketballDrill	74.35	17.20
	RaceHorses	63.50	22.38
ClassD	BlowingBubbles	79.40	40.90
	BQSquare	89.63	37.68
ClassE	Vidyo1	92.03	6.20
	Vidyo4	90.58	7.85
Average		81.58	22.03

zero-block, most coefficients are either zero or close to zero in magnitude, which implies that this kind of nonzero-block is actually very close to the zero-block. Thus, the impact of error-detected blocks on the RD performance is limited. When a zero-block is failure-detected, it only results in less time savings but no impact on the RD performance. Experimental results are demonstrated in Sec. 4.1.

3.2 *Gradients of Residuals for the Weakly Similar Case*

The time-saving performance of the proposed scheme in Sec. 3.1 mainly benefits from the zero-blocks, more precisely, the number of blocks that are detected as zero-blocks. However, it is well known that a smaller QP results in fewer zero-blocks. Moreover, some characteristics of video, such as content texture and motion activity, also affect the number of zero-blocks. The algorithm in Sec. 3.1 is efficient only when many of zero-blocks are present. An algorithm that can be applied to a variety of encoding conditions is required to enhance the complexity reduction. Thus, the algorithm for another case denoted as weakly similar was developed for the fast TU mode decision. In the weakly similar case, block similarity was compared based on the gradients of residuals among the sub-TUs in the pixel domain.

For an $N \times N$ residual block, the gradient along the x direction or y direction is calculated as follows:

$$G_x(i,j) = [I(i,j) - I(i+1,j)], \tag{13}$$

$$G_y(i,j) = [I(i,j) - I(i,j+1)], \tag{14}$$

where $i, j = 0,1,2\ldots,N-1$, and $I(i,j)$ is the residual pixel value. The diagonal residual gradient G is defined as follows:

$$G(i,j) = |G_x(i,j)| + |G_y(i,j)|. \tag{15}$$

An $N \times N$ block has $(N-1)^2$ diagonal residual gradients, and the largest gradient is chosen to represent this block, expressed as

$$G_s = \max_{(N-1)^2}\{G(i,j)\}. \tag{16}$$

For the current TU, four G_s for its sub-TUs can be obtained using Eq. (16); they are denoted as G_{s1}, G_{s2}, G_{s3}, and G_{s4} (Fig. 4). Block similarity is calculated using sum of the absolute difference between each G_s and its mean, written as

$$G_{\mathrm{AD}} = \sum_{i=1}^{4} |G_{si} - \mathrm{avg}\{G_{si}\}|. \tag{17}$$

Because the current TU and its sub-TUs share the same residuals, the value of G_{AD} not only reflects the similarity among the four sub-TUs, but also reveals the relationship among the current TU and its sub-TUs. A small G_{AD} means that the variation of the gradients is small (i.e., these four sub-TUs are similar). This result implies that this TU tends to select the current TU mode without further splitting. By contrast if G_{AD} is large, the residual gradients are dissimilar, and this TU tends to be partitioned to reduce the distortion in the RD cost function.

A threshold $\mathrm{TH}_{\mathrm{grad}}$ is set to determine the weakly similar case. The weakly similar condition is satisfied: if G_{AD} is smaller than $\mathrm{TH}_{\mathrm{grad}}$, an early TU termination is performed and TU splitting is stopped. Otherwise, an early TU splitting is performed, the current TU coding is skipped, and further splitting is directly performed. Consequently, a larger $\mathrm{TH}_{\mathrm{grad}}$ would lead to more early TU terminations and the partition size of the TU would tend to be larger.

The threshold was determined based on extensive experiments. Several distinct thresholds were set for eight

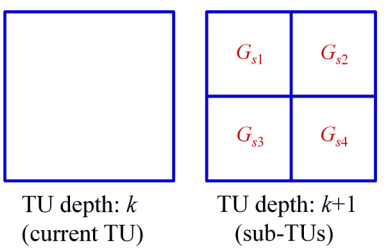

Fig. 4 Illustration of the largest diagonal residual gradients.

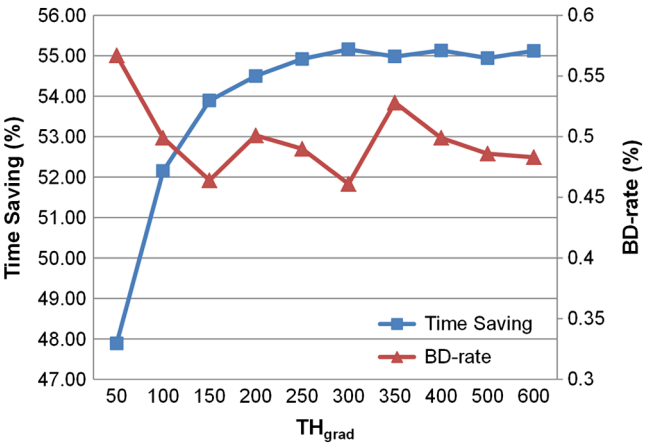

Fig. 5 The average BD-rate and TU time-saving performance for distinct TH_{grad}.

sequences selected from classes C, D, and E. The average Bjontegaard delta (BD)-rate[24] and the TU time-saving performance are depicted in Fig. 5. The threshold was determined to be a favorable trade-off between the RD performance and time saving based on the results of the average BD-rate and time-saving performance. From Fig. 5, the time-saving performance is saturated at ~55% even if the TH_{grad} increases. The threshold was, thus, set to be 300 in the proposed algorithm.

3.3 *Proposed Overall Algorithm*

In the paper, the TU block is classified into the strongly similar case or weakly similar case. The strongly similar case is for zero-blocks, and the weakly similar case is for the variation of gradient values. The flow chart of the proposed algorithm is displayed in Fig. 6. The first step is to check the strongly similar condition. If SAR is smaller than TH_{ZB}, an early TU termination is performed, and the following TU depths are all skipped. If the strongly similar condition is not satisfied, the second step of the weakly similar condition is checked. An early TU splitting is performed if G_{AD} is greater than TH_{grad} (i.e., the current TU and its sub-TUs are dissimilar). The current TU depth is skipped, and the TU is directly split into four sub-TUs. If G_{AD} is smaller than TH_{grad}, an early TU termination is performed because the current TU and four sub-TUs are similar. The weakly similar condition is checked only when the TU size is $>8\times8$ because of RD performance considerations.

4 Experimental Results

Experiments were first designed to test the proposed methods, the strongly similar case, and the weakly similar case. Subsequently, the overall system performance was compared with Choi's algorithm.[19] All the algorithms were implemented on HM14.0. The test platform was a PC with an Intel i7-2600 3.4 GHz CPU, 8 GB RAM, and Windows 7 operating system. The simulation environment for the overall performance evaluation is listed in Table 4. Other settings were the same as the common test conditions[25] used in JCT-VC meetings. The coding performance was measured regarding the BD-rate (BDBR) and average TU encoding time saving as defined in Eq. (18).

$$\Delta T_{TU} = \frac{1}{4}\sum_{i=1}^{4}\frac{\mathrm{TUTime}_{\mathrm{proposed}}(QP_i) - \mathrm{TUTime}_{\mathrm{HM}}(QP_i)}{\mathrm{TUTime}_{\mathrm{HM}}(QP_i)} \times 100(\%). \quad (18)$$

4.1 *Results of Zero-Block Detection Combined with Zero-Block Inheritance for the Strongly Similar Case*

The results for the strongly similar case are shown in Table 5. The BD-rate was nearly lossless in both configurations. The time saving in class E was much larger than that in other classes, because these two sequences exhibited low motion

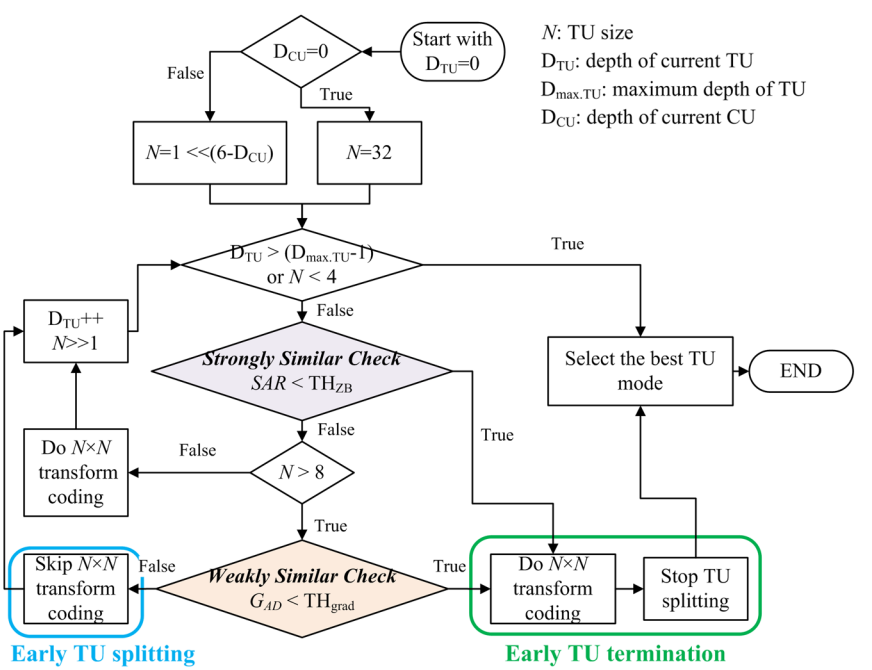

Fig. 6 Flow chart of the proposed algorithm.

Table 4 Simulation environment.

HM14.0		
Sequence	ClassB (1920 × 1080)	BQTerrace, BasketballDrive (excluded in Secs. 4.1 and 4.2)
	ClassC (832 × 480)	BasketballDrill, RaceHorses
	ClassD (416 × 240)	BlowingBubbles, BQSquare, BasketballPass, RaceHorses
	ClassE (1280 × 720)	Vidyo1,Vidyo4
Configuration	Low delay P main (LD_P), Random access main (RA)	
MotionSearch	TZ search	
SearchRange	64	
FramesToBeEncoded	100	
QP	22, 27, 32, 37	
GOPSize	4 (LD_P), 8 (RA)	

and homogenous regions, resulting in a high proportion of zero-blocks. By contrast, this scheme achieved a less effective time saving for sequences with a complex texture and fast motion, such as RaceHorses and BlowingBubbles, than that of other sequences. Compared with the low delay P main (LD_P) configuration, the random access main (RA) configuration can save more time, because the B-slices in the RA configuration achieve a more effective prediction and result in more zero-blocks. The time-saving performance for various QPs is displayed in Table 6 and demonstrates that time saving decreases when the QP value decreases. In summary, this scheme attained a more favorable time saving and RD performance for smoother sequences and higher QP settings.

4.2 *Results of Gradients of Residuals for the Weakly Similar Case*

The results for the weakly similar case are displayed in Table 7. Using this scheme, an average time saving of ~55% and a slight BD-rate increase were achieved. The time-saving performance obtained using various QPs is summarized in Table 8. This scheme attained a stable time saving and a slight BD-rate increase at all QPs because a fixed threshold was adopted and the residual was nearly irrelevant to the QPs.

4.3 *Results of the Overall Algorithm*

Finally, the proposed overall algorithm was compared with Choi's algorithm,[19] and the results are presented in Tables 9

Table 5 BDBR and time saving for the strongly similar case.

Sequence		LD_P		RA	
		BDBR (%)	ΔT_{TU} (%)	BDBR (%)	ΔT_{TU} (%)
ClassC	BasketballDrill	0.23	−26.43	0.01	−30.89
	RaceHorses	0.01	−15.46	0.03	−18.56
ClassD	BQSquare	0.00	−19.00	0.13	−27.43
	BlowingBubbles	0.09	−16.76	0.16	−23.52
	BasketballPass	0.16	−27.53	0.15	−33.43
	RaceHorses	0.09	−14.78	0.00	−18.46
ClassE	Vidyo1	0.22	−53.81	0.14	−60.27
	Vidyo4	0.17	−47.86	−0.06	−61.09
Average		**0.12**	**−27.70**	**0.07**	**−34.21**

Table 6 Time saving of the strongly similar case for various QPs.

	ClassC_BasketballDrill		ClassD_BQSquare		ClassE_Vidyo1	
QP	LD_P (%)	RA (%)	LD_P (%)	RA (%)	LD_P (%)	RA (%)
22	−8.64	−11.66	−2.89	−7.27	−33.07	−43.34
27	−17.38	−21.58	−9.38	−15.22	−52.14	−59.77
32	−30.44	−38.82	−23.11	−34.09	−62.15	−66.81
37	−49.28	−51.48	−40.62	−53.13	−67.89	−71.17

and 10. The total time saving ΔT is also presented in these two tables, and ΔT is defined in Eq. (19). Both algorithms exhibited a favorable RD performance and achieved a considerable TU time saving. Both algorithms achieved excellent time savings for low motion sequences, such as Vidyo1 and Vidyo4, because these sequences exhibit more homogeneous areas resulting in more zero-blocks. However, Choi's algorithm[19] exhibited ineffective time saving on the sequences with high motions and complex textures, such as BlowingBubbles and RaceHorses. The time saving of the reference algorithm[19] was substantially affected by the video content and QP value. On average, the proposed algorithm provided an additional 23.5% TU time saving compared to that in the literature.[19] A total time saving of ∼17% was achieved using the proposed algorithm.

Table 7 BDBR and time saving for the weakly similar case.

		LD_P		RA	
Sequence		BDBR (%)	ΔT_{TU} (%)	BDBR (%)	ΔT_{TU} (%)
ClassC	BasketballDrill	0.47	−54.83	0.24	−54.31
	RaceHorses	0.52	−56.13	0.82	−56.15
ClassD	BQSquare	0.56	−55.64	0.43	−53.87
	BlowingBubbles	0.38	−56.03	0.57	−55.10
	BasketballPass	0.44	−56.91	0.45	−55.76
	RaceHorses	0.39	−55.90	0.72	−57.09
ClassE	Vidyo1	0.57	−52.33	0.32	−52.63
	Vidyo4	0.37	−53.52	0.27	−59.62
Average		**0.46**	**−55.16**	**0.48**	**−55.56**

Table 8 Time saving of the weakly similar case for various QPs.

	ClassC_BasketballDrill		ClassD_BQSquare		ClassE_Vidyo1	
QP	LD_P (%)	RA (%)	LD_P (%)	RA (%)	LD_P (%)	RA (%)
22	−55.61	−54.83	−55.34	−54.82	−54.04	−53.04
27	−55.46	−54.62	−56.48	−55.45	−51.98	−52.74
32	−53.33	−54.33	−55.19	−50.86	−52.47	−51.54
37	−54.92	−53.46	−55.53	−54.33	−50.81	−53.19

Table 9 Performance of the overall algorithm for LD_P configuration.

Sequence		Ref. 19			Proposed		
		BDBR (%)	ΔT_{TU} (%)	ΔT (%)	BDBR (%)	ΔT_{TU} (%)	ΔT (%)
ClassB 1920 × 1080	BasketballDrive	0.22	−36.87	−8.58	0.53	−62.93	−14.84
	BQTerrace	0.25	−41.93	−11.03	0.38	−63.57	−18.49
ClassC 832 × 480	BasketballDrill	0.44	−37.75	−10.43	0.43	−63.14	−17.77
	RaceHorses	0.42	−22.55	−5.82	0.52	−61.00	−16.46
ClassD 416 × 240	BQSquare	0.52	−38.39	−13.54	0.51	−62.50	−21.97
	BlowingBubbles	0.49	−32.69	−9.95	0.50	−61.99	−19.52
	BasketballPass	0.58	−33.89	−9.65	0.39	−63.66	−18.20
	RaceHorses	0.46	−21.34	−6.06	0.47	−61.08	−17.53
ClassE 1280 × 720	Vidyo1	0.22	−59.72	−15.17	0.34	−68.67	−17.52
	Vidyo4	0.27	−54.65	−13.11	0.37	−67.90	−16.61
Average		**0.39**	**−37.98**	**−10.33**	**0.44**	**−63.64**	**−17.89**

Table 10 Performance of the overall algorithm for RA configuration.

Sequence		Ref. 19			Proposed		
		BDBR (%)	ΔT_{TU} (%)	ΔT (%)	BDBR (%)	ΔT_{TU} (%)	ΔT (%)
ClassB 1920 × 1080	BasketballDrive	0.27	−41.30	−8.65	0.69	−64.04	−13.62
	BQTerrace	0.31	−49.17	−10.44	0.74	−64.56	−14.06
ClassC 832 × 480	BasketballDrill	0.08	−40.14	−9.98	0.29	−63.82	−15.75
	RaceHorses	0.50	−27.84	−6.47	0.84	−61.46	−14.98
ClassD 416 × 240	BQSquare	0.21	−45.91	−11.70	0.54	−64.10	−16.33
	BlowingBubbles	0.37	−37.50	−9.20	0.52	−62.41	−16.01
	BasketballPass	0.30	−38.82	−9.51	0.54	−64.19	−15.88
	RaceHorses	0.39	−27.71	−6.91	0.72	−62.03	−15.83
ClassE 1280 × 720	Vidyo1	0.16	−63.63	−13.48	0.35	−70.83	−14.74
	Vidyo4	0.04	−65.57	−11.63	0.22	−73.77	−13.95
Average		**0.26**	**−43.76**	**−9.80**	**0.55**	**−65.12**	**−15.11**

$$\Delta \mathrm{T} = \frac{1}{4}\sum_{i=1}^{4}\frac{\mathrm{EncTime}_{\mathrm{proposed}}(QP_i) - \mathrm{EncTime}_{\mathrm{HM}}(QP_i)}{\mathrm{EncTime}_{\mathrm{HM}}(QP_i)} \times 100(\%). \quad (19)$$

5 Conclusions

In this study, a fast TU mode decision algorithm for HEVC interprediction was proposed to skip TU candidate modes, using the information that can be derived before performing

transform coding. The relationship among the sub-TUs is classified into two cases, namely strongly similar and weakly similar cases. Zero-block detection combined with ZBI is used for the strongly similar case check to decide whether to terminate early at the current TU depth, and this scheme substantially reduces the TU mode candidates at a high QP. The similarity of the residual gradient is used for the weakly similar case check to skip the current TU mode or stop the TU splitting process. This scheme provides a steady time-saving performance at each QP. The proposed algorithm decreases the TU encoding time by ~64% on average, with an average 0.5% BD-rate loss compared with HM14.0. A substantial time-saving improvement is achieved by the proposed algorithm when compared with that of the reference study[19]. Moreover, the proposed fast TU algorithm can easily be integrated with existing fast CU/PU algorithms to further decrease the encoding complexity. In other words, the proposed method can improve the system performance of HEVC.

References

1. JCT-VC, "HEVC draft and test model editing (AHG2)," presented at *13th JCTVC Meeting*, ISO/IEC MPEG and ITU-T VCEG, Incheon, Korea, JCTVC-M0002 (April 2013).
2. International Telecommunication Union, *Recommendation ITU-T H.264: Advanced Video Coding for Generic Audiovisual Services*, ITU-T, Geneva, Switzerland (2003).
3. L. Shen et al., "An effective CU size decision method for HEVC encoders," *IEEE Trans. Multimedia* **15**(2), 465–470 (2013).
4. Z. Y. Chen, H. Y. Chen, and P. C. Chang, "An efficient fast CU depth and PU mode decision algorithm for HEVC," *Lec. Notes Electr. Eng.* **278**, 153–163 (2014).
5. G. Y. Zhong et al., "Fast inter-mode decision algorithm for high-efficiency video coding based on similarity of coding unit segmentation and partition mode between two temporally adjacent frames," *J. Electron. Imaging* **22**(2), 023025 (2013).
6. J. H. Lee et al., "Novel fast PU decision algorithm for the HEVC video standard," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 1982–1985, IEEE, Melbourne, Australia (2013).
7. T. Sotetsumoto, T. Song, and T. Shimamoto, "Low complexity algorithm for sub-pixel motion estimation of HEVC," in *Proc. IEEE Int. Conf. on Signal Processing, Communication and Computing*, pp. 1–4, IEEE, Kunming, China (2013).
8. S. Wang et al., "Fast multi reference frame motion estimation for high efficiency video coding," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 2005–2009, IEEE, Melbourne, Australia (2013).
9. L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. Consum. Electron.* **59**(1), 207–213 (2013).
10. M. Zhang, C. Zhao, and J. Xu, "An adaptive fast intra mode decision in HEVC," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 221–224, IEEE, Orlando, Florida (2012).
11. T. L. da Silva, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC intra prediction mode decision based on EDGE direction information," in *Proc. of the 20th European Signal Processing Conf.*, pp. 1214–1218, IEEE, Bucharest, Romania (2012).
12. G. Corrêa et al., "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.* **57**(4), 1866–1874 (2011).
13. G. Corrêa et al., "Coding tree depth estimation for complexity reduction of HEVC," in *Proc. Data Compression Conf.*, pp. 43–52, IEEE, Snowbird, Utah (2013).
14. M. Grellert et al., "An adaptive workload management scheme for HEVC," in *Proc. Int. Conf. on Image Processing*, pp. 1850–1854, IEEE, Melbourne, Australia (2013).
15. F. Bossen et al., "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1685–1696 (2012).
16. Y. Cheng et al., "Motion search method based on zero-block detection in H.264/AVC," in *Proc. 8th Int. Conf. on Computer Supported Cooperative Work in Design*, Vol. 2, pp. 739–743, IEEE, Xiamen, China (2004).
17. W. He and M. Xiao, "Research on zero-block detection threshold in H.264/AVC," in *Proc. 2nd Int. Conf. on Pervasive Computing and Applications*, pp. 557–561, IEEE, Birmingham, UK (2007).
18. S. W. Teng, H. M. Hang, and Y. F. Chen, "Fast mode decision algorithm for residual quadtree coding in HEVC," in *Proc. Visual Communications and Image Processing*, pp. 1–4, IEEE, Tainan, Taiwan (2011).
19. K. Choi and E. S. Jang, "Early TU decision method for fast video encoding in high efficiency video coding," *Electron. Lett.* **48**(12), 689–691 (2012).
20. J. Kang, H. Choi, and J. G. Kim, "Fast transform unit decision for HEVC," in *Proc. IEEE Int. Congress on Image and Signal Processing*, pp. 26–30, IEEE, Hangzhou, China (2013).
21. Y. Zhang et al., "Fast residual quad-tree coding for the emerging high efficiency video coding standard," *China Commun.* **10**(10), 155–166 (2013).
22. Y. Shi, Z. Gao, and X. Zhang, "Early TU split termination in HEVC based on quasi-zero-block," in *Proc. 3rd Int. Conf. on Electric and Electronics*, pp. 450–454, Atlantis Press, Hong Kong, China (2013).
23. JCT-VC, "High efficiency video coding (HEVC) test model 14 (HM 14) encoder description," presented at *16th JCTVC Meeting*, ISO/IEC MPEG and ITU-T VCEG, San Jose, California, JCTVC–P1002 (2014).
24. G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," presented at *ITU-T Q.6/SG16 VCEG 13th Meeting*, Document VCEG–M33, ITU-T VCEG (2001).
25. JCT-VC, "Common test conditions and software reference configurations," presented at *12th JCTVC Meeting*, ISO/IEC MPEG and ITU-T VCEG, Geneva, CH, JCTVC–L1100 (2013).

Zong-Yi Chen received his BS degree in electrical engineering and his MS degree in communication engineering from National Central University (NCU), Taiwan, in 2005 and 2007, respectively. He is currently pursuing his PhD degree at the Video-Audio Processing Laboratory in the Department of Communication Engineering at NCU, Taiwan. His research interests include video/image processing and video compression.

Jiunn-Tsair Fang received his BS degree in physics from National Taiwan University in 1987 and his PhD degree in electrical engineering from National Chung-Cheng University, Taiwan, in 2004. Currently, he is an assistant professor in the Department of Electronic Engineering at Ming Chuan University, Taiwan. His research interest includes video/image coding, channel coding, and joint source and channel coding.

Chung-Shian Chiang received his BS degree in electrical engineering and his MS degree in communication engineering from Tamkang University and NCU, Taiwan, in 2011 and 2013, respectively. His research interest is video coding. He joined ELAN Microelectronic Corp. in 2013, and his work centered on touch pad algorithm development.

Pao-Chi Chang received his BS and MS degrees from National Chiao Tung University, Taiwan, in 1977 and 1979, respectively, and his PhD degree from Stanford University, California, in 1986, all in electrical engineering. In 1993, he joined the faculty of NCU, Taiwan, where he is presently a professor in the Department of Communication Engineering. His main research interests include speech/audio coding, video/image compression, scalable coding, error resilient coding, digital watermarking, and data hiding.