Speeding Up the Decisions of Quad-Tree Structures and Coding Modes for HEVC Coding Units

Shen-Chuan Tai, Chia-Ying Chang, Bo-Jhih Chen, and Jui-Feng Hu

Institute of Computer and Communication Engineering, Department of Electrical Engineering
National Cheng Kung University
`sctai@mail.ncku.edu.tw`

Abstract. High Efficiency Video Coding (HEVC) is being developed by the joint development of ISO/IEC MPEG and ITU-T Video Coding Experts Group (VCEG) and is expected to be a popular next-generation video codec in the future. HEVC can provide higher compression ratio compared to H.264/AVC standard; however, the coding complexity is dramatically increased as well. In this thesis, a fast algorithm for coding unit decision is proposed to reduce the burden of the encoding time in HEVC. The proposed algorithm exploits the temporal correlation in the neigh-boring frames of a video sequence to avoid the unnecessary examinations on CU quad-trees. In addition, based on an adaptive threshold, the best prediction mode is early determined to SKIP mode for reducing the exhaustive evaluations at prediction stage. The performance of the proposed algorithm is verified through the test model for HEVC, HM 5.0. The experimental results show that the proposed algorithm can averagely achieve about 27%, 33%, 20%, and 21% total time encoding time reduction under Low-Delay High Efficiency, Low-Delay Low Complexity, Random-Access High Efficiency, and Random-Access Low Complexity configurations respectively with a negligible degradation of coding performance.

The rest of this thesis is organized as follows. Section 1 gives a brief introduction to the HEVC encoder, includes overview of HEVC coding standard. Simultaneously, some previously proposed methods for fast CU decision are also investigated in this chapter. Section 2 proposes a new early termination algorithm for CU decision. Section 3 demonstrates the experimental results verified through the test model for HEVC, HM 5.0 [4]. Section 4 concludes the studies presented in this thesis.

Keywords: High Efficiency Video Coding, Coding Unit, Quad-Tree Structure, Coding Mode, Fast Algorithm.

1 Introduction

1.1 Overview of Video Compression

In recent years, many video compression standards have been proposed, such as MPEG-2/-4 [1], H.261/H.263 and H.264/AVC [2]. Some of these standards produce huge commercial interest and gain popular acceptance in the marketplace. For example, the success of digital TV and Digital Versatile Disk (DVD) is exactly based upon

J.-S. Pan et al. (Eds.): *Advances in Intelligent Systems & Applications*, SIST 21, pp. 393–401.
DOI: 10.1007/978-3-642-35473-1_40

MPEG-2. After the successful experience of MPEG-2, Joint Video Team (JVT), a collaborative group of ITU-T Video Coding Experts Group (VCEG) and Moving Picture Experts Group (MPEG), jointly proceeded to develop a new compression standard known as H.264/AVC. It is reported that H.264/AVC provides significantly higher performance in both visual quality and data compression than MPEG-2. Nowadays, a large number of video applications have been dominated by H.264/AVC. These applications utilize highly integrated semiconductor solutions to reduce costs and exhibit the benefit of high-efficiency compression and high-speed decompression in H.264/AVC.

1.2 Overview of HEVC

The encoding layer of HEVC is still based on traditional approach as founded in previous standard designs, including block-based motion-compensated prediction, spatial redundancy prediction, 2D transformation of residual difference signals, and adaptive entropy coding. HEVC integrates many efficient coding tools [9] and provides higher coding performance than earlier video coding standards such as H.264/AVC. The general structure of the HEVC encoder is similar to H.264/AVC. However, there are a number of distinguishing features in HEVC.

Different from previous standards, large sized blocks with flexible structure of quad-tree are applied in HEVC. For this, the HEVC draft specification [6] adopts variable sized coding units (CUs), which define a sub-square region in a frame. HEVC replaces the macroblock scheme as known in previous video coding standards by CU-based quad-tree structure. The encoding process recursively investigates the CUs on the quad-tree. Each CU contains one or several variable sized prediction units (PUs) and transform units (TUs).

The following sub-sections present some specific features in HEVC mentioned above.

Coding Unit (CU)

Coding unit (CU) is considered to be the fundamental processing unit just as macroblock in H.264/AVC. CUs are restricted to be square in shape but conserve the characteristic of variable sizes. CUs vary in size with a wide range corresponding to the depth of the CU quad-tree. A CU with the largest size is called Largest Coding Unit (LCU) and the Smallest Coding Unit (SCU) is defined from the size of the LCU and the maximum depth of the CU quad-tree.

Prediction Unit (PU)

Prediction Unit (PU) is the basic unit used for carrying information related to prediction stage. Each CU on a quad-tree enters its own prediction stage during the encoding process. PUs are not restricted to be square in shape in order to facilitate the partitions which match the boundaries of real objects in video frames.

Transform Unit (TU)

Transform Unit (TU) is used for the transform and quantization of residual signals resulting from prediction stage. TUs must be smaller than or equal to the corresponding CU. Each CU may contain one or more TUs which vary in size from 4×4 up to 32×32.

The TUs in a CU are arranged in quad-tree structure known as residual quad-tree (RQT) just similar to the CU quad-tree. For each CU, a residual quad-tree must be defined, which means that several nested TU quad-trees are embedded in a CU quad-tree. Moreover, under HE configurations, non-square transform (NSQT) is applied for transform step to achieve further coding performance.

2 Proposed Algorithm

Consider previously proposed fast CU decision algorithms. Gweon [8] and Choi [7] early terminate the encoding process within the CU quad-trees based on predefined conditions. Leng [11] and Kim [10] have a lot in common such as referring to spatially neighboring CUs or temporally co-located CUs. The proposed algorithm combines CU quad-tree pruning method (CUQTP) at CU level and early SKIP mode decision at the prediction stage to reduce the computational complexity of the encoding process. At the CU level, some specific depths of a CU quad-tree can be eliminated by referring the coding information of the co-located CUs and CUs adjacent to the co-located CUs. At the prediction stage, the best prediction mode can be early determined to SKIP mode based on adaptive thresholds, and the exhaustive evaluations of INTER modes and INTRA modes is omitted. The overall flowchart of the proposed algorithm is shown in Figure 1. The proposed CU quad-tree pruning method is introduced in Section 2.1, and the proposed early SKIP mode decision is introduced in Section 2.2.

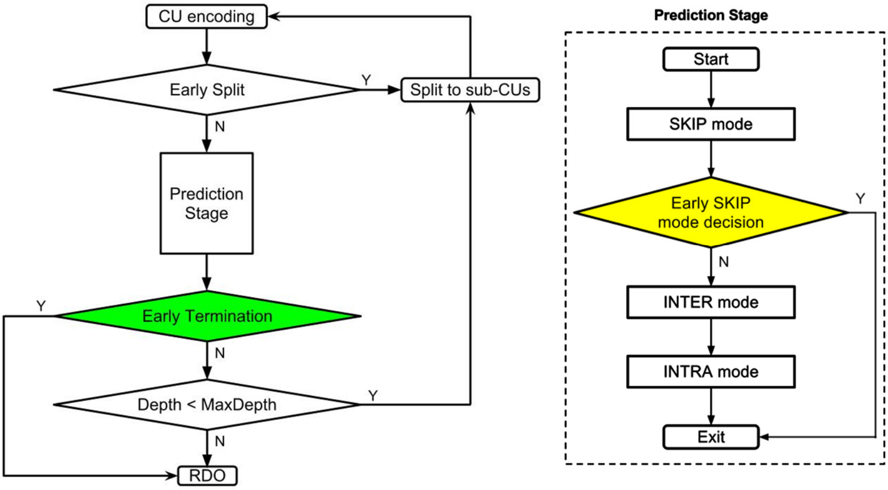
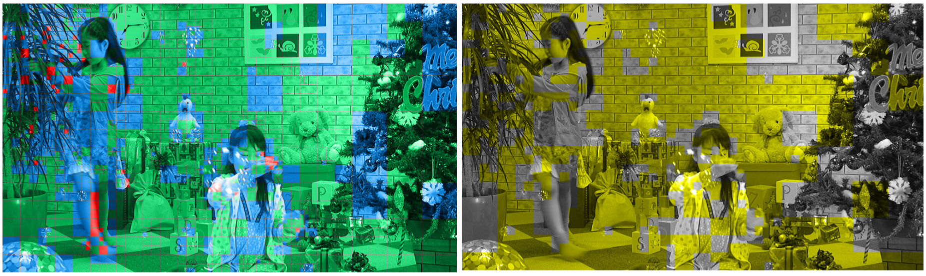


Fig. 1. The overall flowchart of the proposed algorithm

2.1 Coding Unit Quad-Tree Pruning Method

As discussed in Section 1.2, homogeneous textured or low-motion regions tend to be encoded with large CUs. On the other hand, complex textured or high-motion regions tend to be encoded with small CUs. The information of a CU consists of residual signals, motion vectors, and other side information. For several consecutive frames, some features stay the same such as the video resolution, the stationary background and the speed of moving object, and the complex regions and the homogeneous regions stay the same too. Hence, there exists the coding information correlation among consecutive frames [11]. According to these features, the examinations of CUs in specific depth can be passed over by analyzing the coding information in neighboring frames during the encoding process.



(a) The encoding result of the 6th frame (b) The encoding result of the 7th frame

Fig. 2. An example of temporal correlation between neighboring frames in PartScene

Figure 2(a) and Figure 2(b) show two consecutive B-frames in the sequence. The yellow region in Figure 2(b) is encoded with the same size of CUs as the temporally co-located region in Figure 2(b). As shown in the figure, most part of the background and many stationary objects have temporal correlation on the two consecutive frames. Some regions with complex textures or lighting changes are encoded with different size of CUs due to the zoom effect.

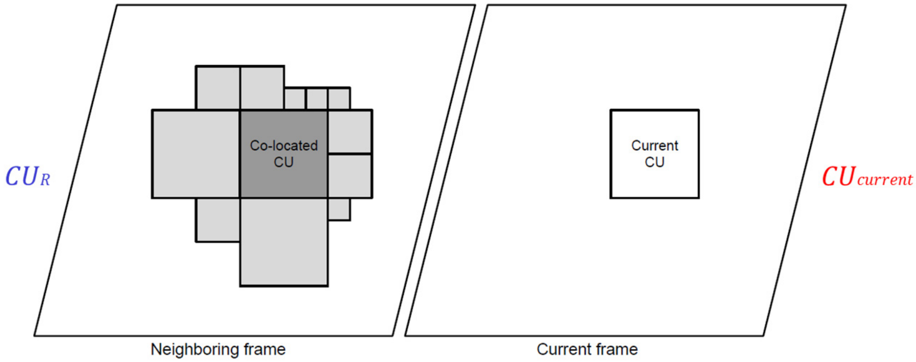
To evaluate the temporal correlation between frame t and frame t − 1, the Similarity - Ratio^t (SR^t) is defined as:

$$SR^{t}=\frac{\sum_{N}\left(1\left|depth\left(SCU_{(i,j)}^{t}\right)=\right|depth\left(SCU_{(i,j)}^{t-1}\right)\right)}{N} \tag{1}$$

where $\mathrm{SCU}^{t}_{(i,j)}$ is the basic statistical unit which allocates a SCU-sized block at (i , j) position, depth(.) is the depth (i.e., the size) of the encoded CU onto the block, t is the display order of frames, and N is the total number of SCU-sized blocks in a frame. SR^t is the area ratio of frame t that encoded with the same size of CUs as the co-located region in the previous frame.

The proposed CU quad-tree pruning method relies on the temporal correlation of CU quad-tree. A cluster of CUs (CU$_R$), which consists of the temporally co-located CU and CUs adjacent to the co-located CU in the encoded neighboring frame. Figure 3 shows the relationship between the current CU (CU$_{current}$) and the CU$_R$. There are uncertain numbers of CU$_s$ in the CU$_R$. Before the CU$_{current}$ enters the prediction stage, the coding information of the CU$_R$ is used for determining whether to perform the prediction stage or not. There are two additional conditions, "Early Split" and "Early Termination", included in the general encoding process.

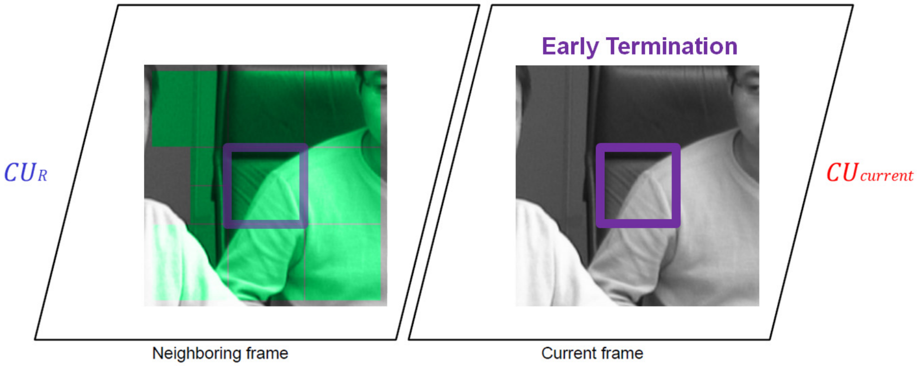
First, the Early-Split condition is defined as Figure 3.



$$depth(CU_x) > depth(CU_{current}) \\ \vee \\ depth(CU_x) = depth(CU_{current}) \;\wedge\; mode(CU_x)! = SKIP, \\ \text{for all } CU_x \in CU_R$$

Fig. 3. Condition for early split

Second, the Early-Termination condition is defined as Figure 4.



$$depth(CU_x) < depth(CU_{current}) \;\vee\; mode(CU_x) = SKIP, \\ \text{for all } CU_x \in CU_R$$

Fig. 4. Condition for early termination

2.2 Early Skip Mode Decision

As aforementioned, the encoder computes the R-D costs of all possible inter prediction modes and intra prediction modes to decide the best prediction mode. Since each of them entails high computational complexity, it is very desirable if the encoder can decide the best prediction mode at the earliest possible stage without evaluating all possible prediction modes exhaustively.

3 Experimental Results

The proposed algorithm was implemented in the test model HM5.0 of HEVC. The test platform is AMD Dual-Core Socket F Opteron 2220 2.8 GHz, 8.0 GB RAM. A group of experiments were carried out on the recommended sequences with quantization parameters 22, 27, 32, and 37. According to the specifications provided in [5], four encoder configurations including Low-Delay High Efficiency (LBHE), Low-Delay Low Complexity (LBLC), Random-Access High Efficiency (RAHE) and Random-Access Low Complexity (RALC) are used to verify the proposed algorithm.

The objective quality of the reconstructed frames of video sequence is evaluated by the peak signal-to-noise ration (PSNR), which is defined as:

$$PSNR = 10\times\log_{10}\frac{255^2}{\frac{1}{M}\sum_{n=1}^{M}(o_n\text{-}r_n)^2} \tag{2}$$

where M is the number of samples, and o_n and r_n are the gray level of the original and reconstructed frames, respectively. On the other hand, another factor influencing overall coding performance is the bit-rate after compressing the video sequence. In our experiment, the coding performance was evaluated based on the ΔBit-Rate, ΔPSNR defined as follows:

$$\Delta\text{Bi-Rate}(\%)=\frac{\text{Bit-rate}_{\text{proposed}}\text{-Bit-rate}_{\text{HM5.0}}}{\text{Bit-rate}_{\text{HM5.0}}}\times 100\% \tag{3}$$

$$\Delta\text{PSNR(dB)=PSNR}_{\text{proposed}}-\text{PSNR}_{\text{HM5.0}} \tag{4}$$

Besides, Bjontegaard [3] integrates the two factors into an objective evaluation indicator know as Bjontegaard delta bit-rate (BDBR). Based on rate-distortion curve fitting, the BDBR represents the average bit-rate difference in percentage over the whole range of PSNR. This evaluation indicator is also applied in our experiment. For the complexity evaluation, the total execution time of the proposed algorithm is assessed in comparison to that of HM 5.0. The time reduction in computational was evaluated based on ΔEncT(%) as follows:

$$\Delta\text{EncT}(\%)=\frac{\text{Time}_{\text{proposed}}-Time_{\text{HM5.0}}}{Time_{\text{HM5.0}}}\times 100\% \tag{5}$$

Thirteen benchmark test sequences are selected to be encoded with four different quantization parameters (QPs: 22, 27, 32, 37) under different encoder settings.

Table 1 shows the detailed information of the test sequences. These sequences have various characteristics including motion types, texture types, and resolutions.

Table 1. Benchmark test sequences

Class	Sequence	Frames	FrameRate
B 1920×1080	Kimono	240	24 fps
	ParkScene	240	24 fps
C 832×480	BasketballDrill	500	50 fps
	BQMall	600	60 fps
D 416 ×240	BasketballPass	500	50 fps
	BQSquare	600	60 fps
	BlowingBubbles	500	50 fps
	RaceHorses	300	30 fps
E 1280× 720	Vidyo1	600	60 fps
	Vidyo3	600	60 fps
	Vidyo4	600	60 fps

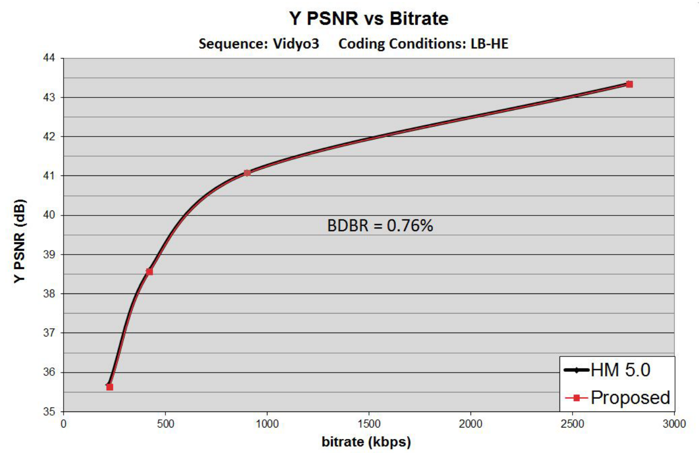
Figure 5 shows the worst cases of R-D curves compared with that of HM 5.0. As shown in the figures, the RD curves almost overlap in each case. The results indicate that the proposed algorithm has almost the same coding performance as that of HM 5.0.

Table 2 summarizes results of the proposed algorithm with the BDBR and average time reduction. It should be noted that positive values represent bit-rate increase (i.e., PSNR degradation). The results of average BDBR are 0.39%, 0.47%, 0.10% and 0.10% for LBHE, LBLC, RAHE, RALC, respectively, and the average time reductions are -27%, -33%, -20% and -22% for the four configurations.

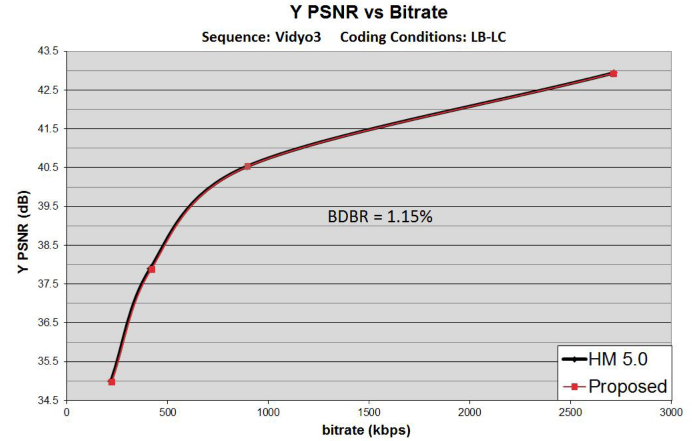
4 Conclusions

In this thesis, a fast algorithm for speeding up the decision of quad-tree structures and coding modes for HEVC coding units is proposed. The CU-based quad-tree structure is responsible for the complexity of the encoding process. The proposed algorithm accelerates the encoding process on two aspects including CU level and prediction stage.

At the CU level, the proposed algorithm prunes the CU quad-tree to a condensed shape based on the temporal correlation between neighboring frames. At prediction stage, the best prediction mode is early determined to SKIP mode based on an adaptive threshold. The main contribution of this thesis is to provide a simple and efficient fast algorithm with very negligible loss of coding performance compared to the original HEVC encoder. On the other hand, the proposed algorithm can be easily combined with the early termination method of mode decision [10] which was adopted in the HM5.0 anchor. In terms of the complexity reduction, the combined algorithm can speed up the encoding process considerably with reasonable degradation of coding performance.



(a) RD curve of Vidyo3 under LBHE



(b) RD curve of Vidyo3 under LBLC

Fig. 5. The R-D performance of the proposed algorithm with Vidyo3

Table 2. Overall experimental results with BDBR (%) and ΔEncT (%)

Class	Sequence	LBHE		LBLC		RAHE		RALC	
		BDBR	ΔEncT	BDBR	ΔEncT	BDBR	ΔEncT	BDBR	ΔEncT
B	Kimono	0.33	-24	0.35	-26	-0.10	-15	0.06	-16
	ParkScene	0.50	-25	0.79	-29	-0.05	-15	0.09	-17
C	BasketballDrill	0.36	-15	0.31	-21	0.07	-10	0.05	-11
	BQMall	0.38	-18	0.50	-25	0.14	-14	0.15	-15
	PartyScene	0.31	-14	0.45	19	0.11	-13	0.11	-13
	RaceHorse	0.31	-14	0.45	19	0.11	-13	0.11	-13
D	BasketballPass	0.21	-14	0.21	-19	0.20	-12	0.02	-12
	BQSquare	0.25	-15	0.30	-21	0.18	-14	0.17	-15

Table 2. (*continued*)

	BlowingBubbles	0.42	-15	0.34	-20	0.22	-12	0.12	-12
	RaceHorse	0.21	-11	0.16	-15	0.24	-12	0.23	-11
E	Vidyo1	0.69	-36	0.90	-43	0.09	-28	0.11	-31
	Vidyo3	0.76	-37	1.15	-43	0.07	-28	0.06	-31
	Vidyo4	0.50	-37	0.44	-44	0.04	-29	0.04	-32
Average		0.39	-27	0.47	-33	0.10	-20	0.10	-21

References

1. Information tecchonology: Generic coding of moving pictures and associated audio information, Std. (1995)
2. Advanced video coding for generic audiovisual services, Std. (2007)
3. Bjøntegaard, G.: Calculation of average PSNR differences between RD-curves, ITU-T VCEG-M33 (2001)
4. Bossen, F.: HM reference software 5.0 (2011), `https://hevc.hhi.fraunhofer.de/svn/svnHEVCSoftware/`
5. Bossen, F.: Common HM test conditions and software reference configurations (January 2012), `http://phenix.int-evry.fr/jct/docenduser/documents/7Geneva/wg11/JCTVC-G1200-v2.zip`
6. Bross, B., Han, W.-J., Sullivan, G.J., Ohm, J.-R., Wiegand, T.: High Efficiency Video Coding (HEVC) text specification Working Draft 5 (February 2012), `http://phenix.int-evry.fr/jct/docenduser/documents/7Geneva/wg11/JCTVC-G1103-v12.zip`
7. Choi, K.H., Jang, E.S.: Coding tree pruning based cu early termination (July 2011), `http://phenix.int-evry.fr/jct/docenduser/documents/6Torino/wg11/JCTVC-F092-v3.zip`
8. Gweon, R.H., Lee, Y.L., Lim, J.Y.: Early termination of cu encoding to reduce hevc complexity (July 2011), `http://phenix.int-evry.fr/jct/docenduser/documents/6Torino/wg11/JCTVC-F045-v1.zip`
9. Jeong, S.Y., Lim, S.C., Lee, H.Y., Kim, J.H., Choi, J.S., Choi, H.C.: Highly efficient video codec for entertainment-quality. ETRI Journal 33, 145–154 (2011)
10. Kim, J.H., Jeong, S.Y., Cho, S.H., Choi, J.S.: Adaptive coding unit early termination algorithm for hevc. In: 2012 IEEE International Conference on Consumer Electronics (ICCE), pp. 261–262 (January 2012)
11. Leng, J., Sun, L., Ikenaga, T., Sakaida, S.: Content based hierarchical fast coding unit decision algorithm for hevc. In: 2011 International Conference on Multimedia and Signal Processing (CMSP), vol. 1, pp. 56–59 (May 2011)