A Fast Depth Determination Algorithm for High Efficiency Video Coding

GAO Yuan Yuan[1, a], WU Xian Yun[2,b], LI Yun Song[3,c] and Liu Kai[4,d]

[1,2,3,4]State Key Laboratory of Integrated Services Networks, Xidian Univ.,Xi'an 710071, China

[a]gyy6654@sina.com, [b]xywu@mail.xidian.edu.cn, [c]ysli@mail.xidian.edu.cn, [d]kailiu@mail.xidian.edu.cn

Keywords: HEVC, Inter prediction, depth determination

Abstract. In this paper, a fast depth partition algorithm for High Efficiency Video Coding (HEVC) is proposed. HEVC improves the coding efficiency by using large coding units while increases the coding complexity evidently. This paper analyzes the depth correlation between adjacent frames to fast determine the depth of current coding unit. The skip mode is also adopted for better performance. Experimental results show that our algorithm can reduce 52% coding complexity compared with original HEVC algorithm. Meanwhile, the average PSNR only decreases 0.09dB and the average coding rate is decreased about 0.58%.

Introduction

Video coding is the process of compressing and decompressing a digital video series, which has been widely used in TV broadcasting, video conference and DVD applications [1]. The High Efficiency Video Coding (HEVC) [2] has emerged with the increasing video resolution and increasing use of parallel processing architectures, and has been established as a new video coding standard at 2010. Compared to the H.264/AVC, HEVC decreases half bit-rate while achieving similar video quality, but coding complexity increases significantly. How to decrease coding complexity and keep the image quality and the bit-rate is new challenge.

In this paper, a fast inter depth determine algorithm is proposed. We analyze the correlation of the coding depth between adjacent frames. Using the previous coding depth information, skips or terminates the depth split in advance, and adopts the skip mode to decrease the coding complexity.

Inter Prediction Analysis

HEVC adopts an adaptive partitioning and prediction quad-tree architecture. The coding frame is split into different Coding Units (CU), and the size of CU is not fixed [3]. The quad-tree structure also makes encoding process more complex. To choose the best LCU partition, it needs to compute the rate-distortion cost of all depths, each depth needs to perform prediction, transform, quantization and entropy coding to decide whether the CU is split or not [4]. Each LCU will get the best quad-tree structure after the quad-tree traversal, as shown in Fig.1.

The lower left side of Fig.1 shows the split CU structure, and the lower right side shows the quad-tree structure. As we can see from Fig.1, 64×64 LCU size needs processing prediction, transform, quantization and entropy coding. When the maximum depth is 3, one LCU needs 64 times CU traversal [5,6]. E.g. BasketballDrill, there are 104 LCUs in one frame, 6656 times CU traversal need to be done in total. And every traversal needs computing the best prediction mode, transform, quantization and entropy coding. The introduction of the more flexible partition mode and bigger coding units increases coding efficiency while increasing the computation sharply. So, it is very important to optimize the CU split [7,8].
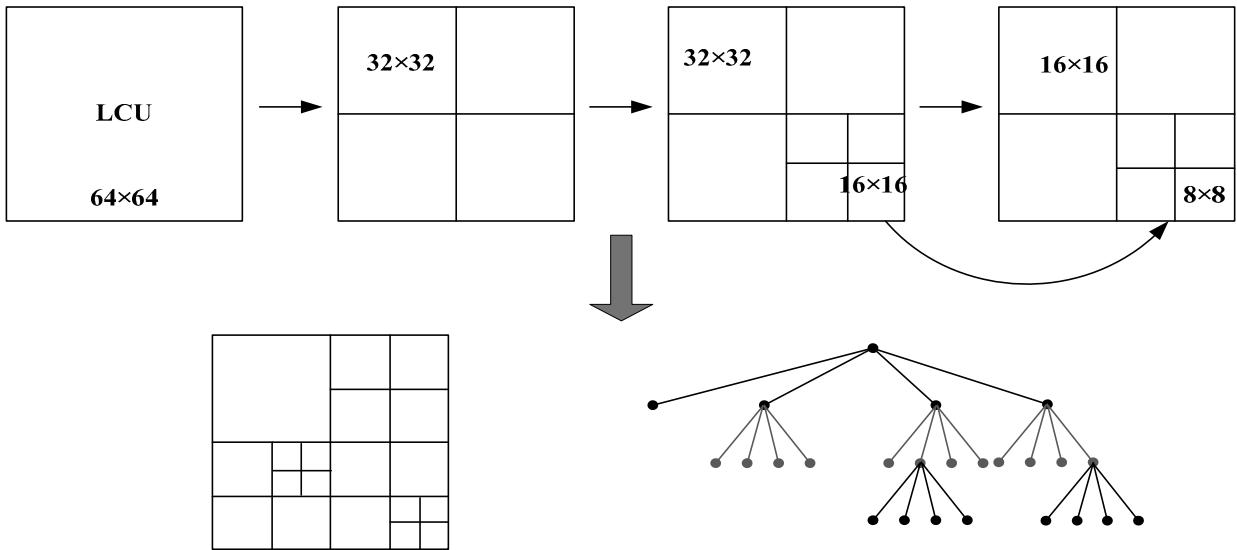
Figure 1 quad-tree traversal of CU partition

A video consists of a series of consecutive frames. Consecutive frames differ little in temporal domain and present strong relativity[9,10]. According to the characteristic and the CU split depth structure, we find two neighbor frames have strong relativity as depicted in Figure 2.
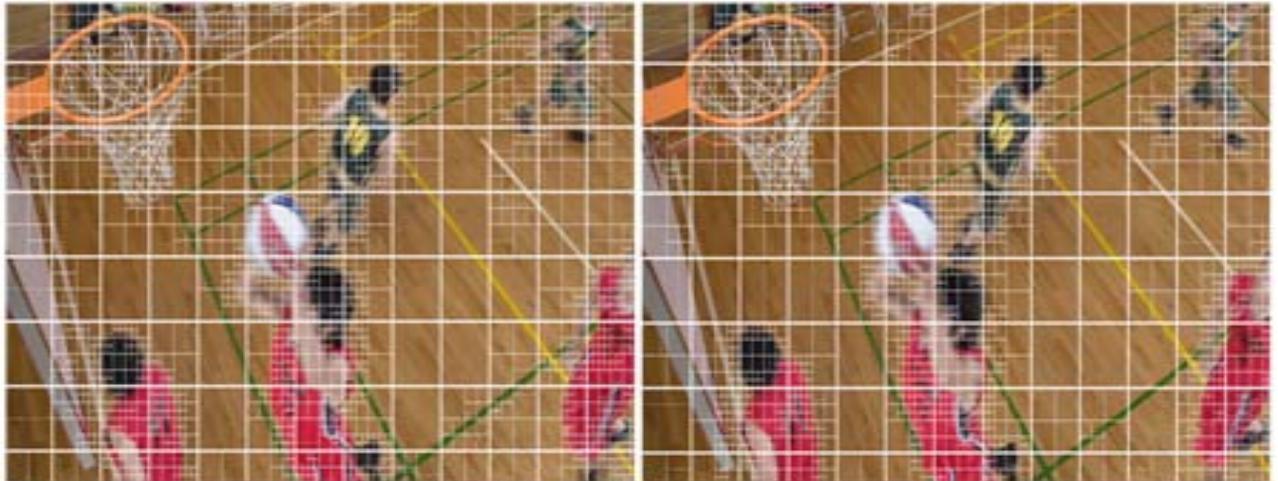
(a) the 22th frame   (b) the 23th frame

Figure 2 the best CU partition mode of consecutive frames

Table1 consecutive frames CU depth correlation

Sequence(resolution)	Ratio of CU partition depth relative[%]
BlowingBubbles (416×240)	90.2
BasketballDrill (832×480)	95
vidyo1 (1280×720)	97.5
ParkScene (1920×1080)	89.2

To validate the correlation of neighbor frames, a series of test is made using BlowingBubbles (416×240), BasketballDrill (832×480), vidyo1 (1080×720) and ParkScene (1920×1080). We calculate the same CU partition depth ratio between neighbor frames, but the ratio is too small. So extend the depth to present depth from minus 1 to plus 1. The result is listed in Table 1. In this

instance, the neighbor frames have strong relativity. So the partition CU depth of previous frame can be the reference CU depth for the present frame to decrease the traversing times.

Fast Depth Determination Algorithm

We analyze the inter prediction algorithm complexity and test standard sequences. The result shows the CU depth between the present frame and its reference frame has strong correlation. Also, the skip mode doesn't need to code the residual, so it can decrease coding and splitting complexity, and decrease coding bit-rate and coding time as well. The PSNR would not decrease clearly. Thus skip mode is adopted to increase the algorithm efficiency.

Using the relativity of CU and reference CU, adopting the algorithm of current CU depth +1, -1 and skip mode decision. The procedure of our algorithmic is listed as following:

Step 1：If current slice is I slice, coding normally, else go to step 2.

Step 2: If the current CU depth is less than the CU depth of same location in previous frame, skip the current coding depth, split four sub-CUs directly, else go to step 3.

Step 3: If the current CU depth is more than the CU depth of the same location in previous frame, just compute the current CU depth and do not split, else go to step 4.

Step 4: Coding current CU normally and keep splitting.

The flow chart of the proposed algorithm is shown in Figure 3
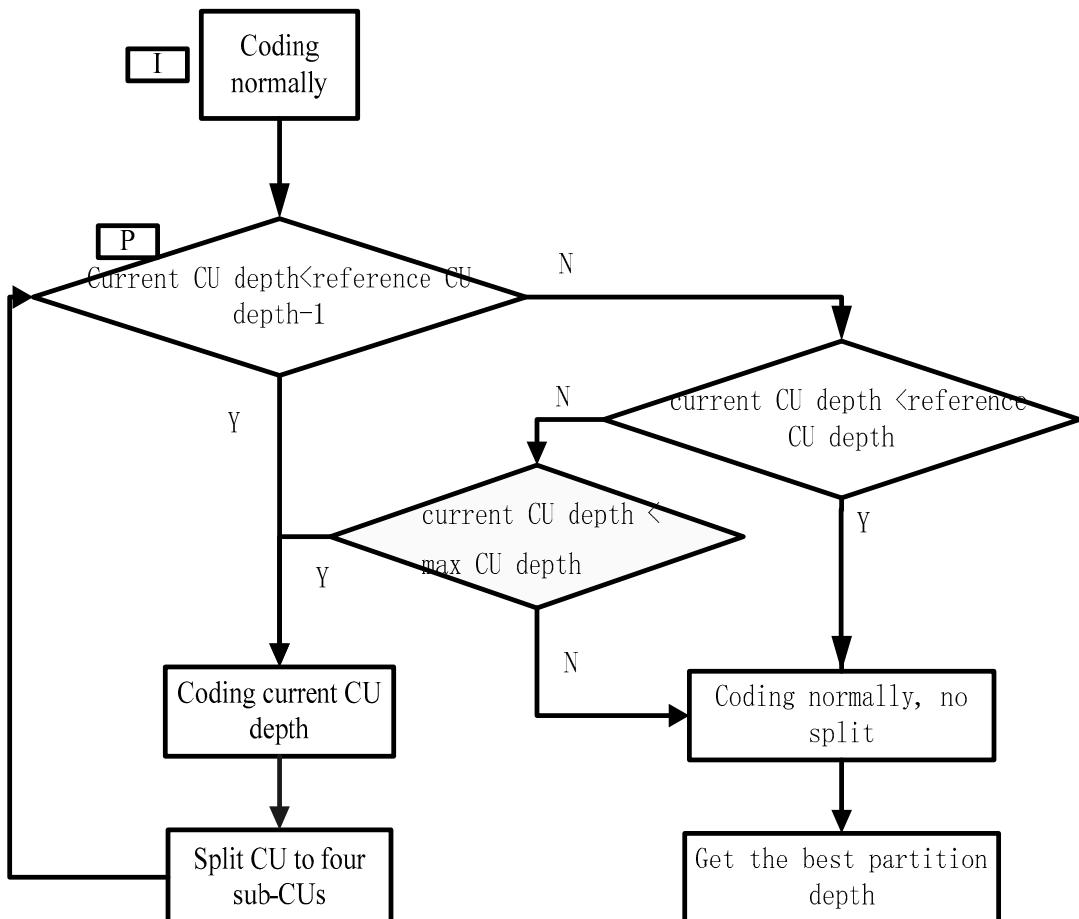
Figure 3 The proposed algorithm

Experimental Results

To verify our algorithm, we use four standard sequences of different resolutions. The proposed method is implemented on HM10.0 reference software [2]. Under the encoder_lowdelay_P_main configuration, 100 frames are encoded with given QP values (22, 27, 32 and 37). The maximum size

of CU is 64*64，the maximum CU depth is 3. All tests use one computer and hardware parameter is Intel(R) Core(TM) 2 Duo CPU, CPU 2.93GHz, 2.00GB RAM. And the four sequences are BlowingBubbles (416×240), BasketballDrill (832×480), Vidyo1 (1280×720) and ParkScenc (1920×1080).

This paper uses ΔBit, $\Delta PSNR$ and $\Delta Time$ to measure the final performance, and the encoding time saving is given by,

$$\Delta Time = \frac{Time_{HM} - Time_{new}}{Time_{HM}} \times 100\% \tag{1}$$

Table 2 shows the experimental results. The average encoding time saved of the proposed algorithm is about 52%. And 0.58% bit-rate increased, while only 0.09dB PSNR loss. The proposed algorithm keeps the image quality with only little PSNR loss. We should notice that the skip mode decreases the bit-rate. And the higher the resolution is, the higher the coding time decrease achieved.

Table2 Result of the proposed algorithm compared with HM encoder

QP	Sequences	ΔBit [%]	$\Delta PSNR$ [dB]	$\Delta Time$ [%]
22	Blowbubbles	0.18	-0.04	36.08
	BasketballDrill	0.37	-0.03	35.24
	Vidyo1	-1.60	-0.05	57.15
	ParkScene	-0.47	-0.05	40.56
27	Blowbubbles	-0.47	-0.09	39.50
	BasketballDrill	-0.28	-0.07	38.20
	Vidyo1	-1.70	-0.08	70.57
	ParkScene	-0.48	-0.09	44.53
32	Blowbubbles	-0.13	-0.13	40.96
	BasketballDrill	-0.44	-0.10	48.33
	Vidyo1	-1.32	-0.09	76.20
	ParkScene	-0.40	-0.11	56.87
37	Blowbubbles	-0.39	-0.13	50.00
	BasketballDrill	-0.75	-0.10	56.74
	Vidyo1	-0.67	-0.08	81.57
	ParkScene	-0.76	-0.12	67.39
Average		-0.58	-0.09	52.49

Conclusions

In this paper, a new fast CU depth partition algorithm is proposed. By analyzing encoded reference frame to determine current CU coding depth to decrease encoding time with slight quality loss. Experimental results show that the average encoding time saving is about 52%, and the bit-rate decrease is 0.58%, while the average PSNR loss is only 0.09dB.

(a)Original algorithm (b)Our algorithm

（PSNR=31.32dB） (PSNR=31.23dB)
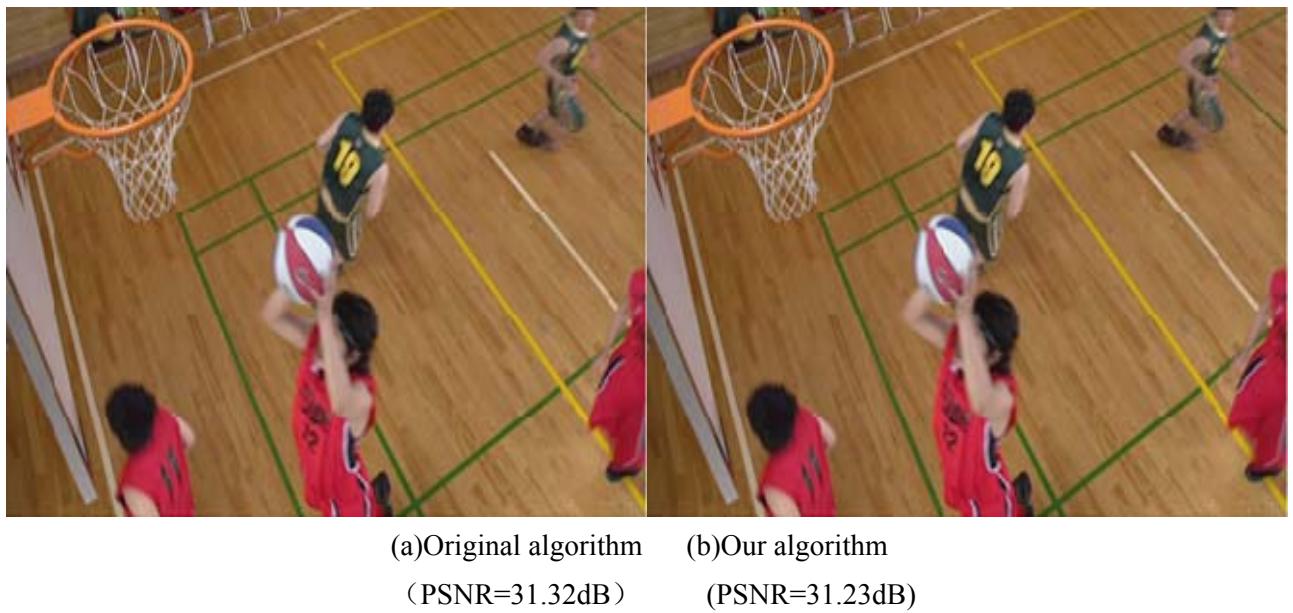
Figure 4 reconstructed 17 frame of BasketBallDrill sequence

Acknowledgements

Reference

[1]. Lin Sun, Oscar C. Au, Cong Zhao, et al. Rate distortion modeling and adaptive rate control scheme for high efficiency video coding (HEVC). Circuits and systems (ISCAS), 2014 IEEE International Symposium on, 2014, 1(5): 1933-1936.

[2]. Sullivan, G. j., Ohm, J. R, Han, W. J, Wiegand, T., Wiegand, T.. Overview of the High Efficiency Video Coding (Hevc) Standard. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1649-1668.

[3]. Xiaotao Hou, Yonglin xue. Fast coding unit partitioning algorithm for HEVC.Consumer Electronics(ICCE), 2014 IEEE International Conference Conference on, 2014, 10(13): 7-10.

[4]. Corrêa G, Assuncao P, Agostini L, et al. Complexity control of high efficiency video encoders for power-constrained devices. Consumer Electronics, IEEE Transactions on, 2011, 57(4): 1866-1874.

[5]. Ohm J, Sullivan G J, Schwarz H, et al. Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC). Circuits and Systems for Video Technology, IEEE Transactions on, 2012, 22(12): 1669-1684.

[6]. Jie J, Baolong G, Wei M, et al. Fast intra coding algorithm using smooth region detection for HEVC. Journal of Xidian University. 2013, 40(3): 194-200.

[7]. Hou X, Xue Y. Fast coding unit partitioning algorithm for HEVC. Consumer Electronics (ICCE), 2014 IEEE International Conference on. IEEE, 2014: 7-10.

[8]. Yoo H M, Suh J W. Fast coding unit decision based on skipping of inter and intra prediction units. Electronics Letters, 2014, 50(10): 750-752.

[9]. Chengyi L, Guowei Teng, Xuli Shi, et al. Fast Intra Prediction algorithm for HEVC. Video Engineering, 2012, 36(21): 4-7.

[10]. Kim I K, Min J, Lee T, et al. Block partitioning structure in the HEVC standard. Circuits and Systems for Video Technology, IEEE Transactions on, 2012, 22(12): 1697-1706