CU splitting on frame boundary for arbitrary resolution video in HEVC

Wei Li[1], Fuzheng Yang[1], Shuai Wan[2] and Guangliang Ren[1]

[1]State Key Laboratory of ISN, Xidian University , Xi'an, 710071, China

[2]School of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710072, China

Email: wli1@mail.xidian.edn.cn

Abstract - A highly flexible quadtree structure has been introduced in the latest high efficient video coding (HEVC) to efficiently represent video content. However, the performance of the quadtre structure would be impaired when the video resolution is not an integral multiply of the largest coding unit (CU). To address this, a new CU splitting method on the frame boundary for arbitrary resolution video is proposed to improve the coding efficiency of HEVC. This method introduces the non-square CU to extend the CU candidate set on the frame boundary when the video resolution is not an integral multiple of the largest CU size. Simulation results show that the proposed method can reduce the coding bit rate by up to 0.76% compared with the original TMuC 0.9 reference software.

1. Introduction

Video coding techniques have been simulated by the rapidly growing demand for video communications. With the increasing popularity of ultra high definition TV, current video coding standards are facing new technical challenges in coding efficiency. Recently, the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), have been actively researching the advanced emerging video coding techniques and targeting to put a major step forward in coding performance. Based on the similar destinations and the previous success in jointly creating H.264/AVC, a Joint Collaborative Team on Video Coding (JCT-VC) was established by ITU-T/VCEG and ISO/IEC/MPEG for joint work in January 2010, which aims to develop the next generation video coding standard called high efficiency video coding (HEVC) [1].

For efficient and flexible representation of video content with various resolutions in HEVC, the video compression framework is designed to be a highly flexible quadtree structure based on a triplet of coding unit (CU), prediction unit (PU), and transform unit (TU) [2]. The CU is defined as a basic unit which is in a square shape and has various sizes. All coding processing except for frame-based loop filtering is performed on the CU basis, including intra/inter prediction, transform, quantization and entropy coding. There are also two special terms defined: the largest coding unit (LCU) and the smallest coding unit (SCU). LCU size and SCU size are limited to values which are a power of 2 and the optimum LCU and SCU are dependent on the video application. The size of CUs in a video frame is limited between the size of the LCU and that of the SCU. PU is a basic unit of inter and intra prediction and there are multiple PU modes in a single CU. TU is employed for large integer transformation, with sizes of 16x16 and 32x32 supported.

With the above definition, a video frame in HEVC would be split into non-overlapped LCUs firstly, and then coded based on the CUs in a recursive tree structure within an LCU. In general, the LCU size is larger than 16x16 for flexible representation, as recommended in HEVC. However, it is possible that the video resolution is not an integral multiple of the LCU size. In this case, the CU modes within the LCU on the frame boundary could be limited and not efficient for coding performance. Thus, in this paper, a new CU splitting method on the frame boundary is proposed for arbitrary resolution video in HEVC and the performance evaluation shows about 0.76% bit rate improvement compared with the reference method under the same coding quality.

The remainder of this paper is organized as follows. Section 2 analyzes the existing problem concretely and presents the proposed method in detail. The experimental results and comparisons are described in Section 3. Section 4 concludes the paper.

2. Proposed cu splitting method on frame boundary

2.1 Problem statement

The concept of the CU is similar to the macroblock and sbu-macroblock in H.264/AVC, where a CU can have various sizes. Fig. 1 shows a quad-tree based CU structure in a picture. The picture is split into a sequence of non-overlapped LCUs and a LCU can be divided into four CUs. Each CU can be further recursively divided into up to four CUs limited by the allowed SCU. For example, if the size of LCU is equal to 64 and that of SCU is 8, then 4 kinds of the CU size are possible: 64x64, 32x32, 16x16 and 8x8. Coupled with a CU, the PU is defined only for the CU as a leaf node. Fig. 1 shows different PU splitting modes for a CU of size 2Nx2N. The prediction method is specified by the prediction mode and the PU splitting. The PU for the intra modes has 2 different possible splittings: 2Nx2N and NxN. The PU for the inter modes has 8 different possible splittings: 4 symmetric splittings (2Nx2N, 2NxN, Nx2N, NxN) and 4 asymmetric splittings (2NxnU, 2NxnD, nLx2N and nRx2N). A skipped PU can only be 2Nx2N. For example, if the size of CU is 64x64, then both 64x64 and 32x32 PUs for intra are possible. For inter prediction, 64x64, 64x32, 32x64, 32x32, 64x16, 64x48, 16x64 and 48x64 are possible. In addition to the CU and PU definitions, the TU for transform and quantization is defined separately. It should be noted that size of a TU may be larger than a PU. It is beneficial for coding semi-random residuals generated by complex texture areas where neighboring partitions share common signal characteristics, which can exploit the better energy compaction of larger transform [3].
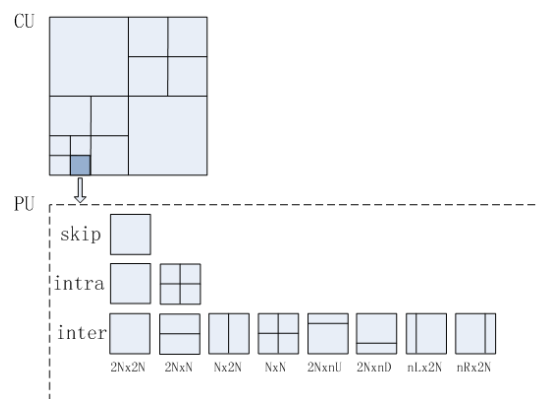

Fig. 1 An example of CU and PU.

It is seen that the new coding framework based on the triplet of CU, PU and TU is superior to the predecessors, representing video content more efficiently. Nevertheless, it is noticeable that the performance may be impaired when the video resolution is not an integral multiple of the LCU size. According to the quad-tree based CU structure, Fig. 2 shows the CU modes of a coded frame of the "Foreman" sequence of resolution 176x144, with the LCU size of 64x64. This frame size is not an integral multiple of the LCU size and the sizes of LCUs on the frame boundary only included actual pixels are 48x64, 64x16 and 48x16. It can be observed that there are numerous small CUs within the LCU on the frame boundary. Obviously, these are not the optimal CU modes, due to the unnecessary cost of motion information and discrete cosine transform (DCT) for the small CUs on the frame boundary. At the same time, the coding mode decision is hardly adapted to video content. Consequently, it should be noticed that the CU modes within the LCU on the frame boundary could be rather limited and not efficient for coding performance when the video resolution is not an integral multiple of the LCU size. Accordingly, a new CU splitting method on the frame boundary is proposed for arbitrary resolution video in HEVC in this paper, as described in rest of this section.
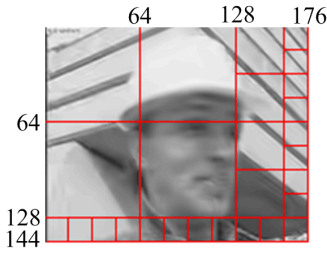
Fig. 2 CU modes for "Foreman" using the TMuC 0.9 software.

2.2 The proposed method

The general idea is to extend the CU candidate set on the frame boundary by introducing non-square CU modes when the video resolution is not an integral multiple of the LCU size. The proposed method is illustrated as follows. For example, the rectangular blocks of 64x16, 32x16 are added into the CU candidate set for saving the bit rate of motion and transform information, where the LCU is size of 64x64 and the block with actual pixel values on frame boundary is size of 64x16.

Assume that the video resolution is not an integral multiple of the LCU size, and the LCU is size of $L \times L$. Since the LCUs on the frame boundary only partially cover frame pixels, it is supposed that the block with actual pixel values within a LCU on the frame boundary is size of $M \times N$.

To begin with, determine the value of L_1, which is generated by (1):

$$L_1 = 2^n, \quad n = Max(3, \lceil Max(\lg M, \lg N) \rceil) \tag{1}$$

where n is an integer and larger than 2. L_1 depends on the video resolution and the LCTB size, and is related to the CU structure within the LCU on the frame boundary, as shown in Fig. 3.
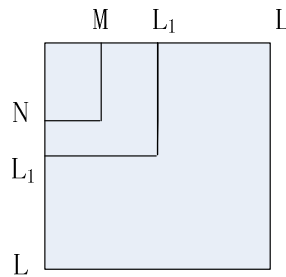

Fig. 3 The splitting framework for CU structure.

Then, determine the CU candidate set as: a) add the block mode of $M \times N$ into the CU candidate set as one of CU extension modes; b) split the block of $L_1 \times L_1$ into four sub-units of equal size. Simultaneously, the block of $M \times N$ may be divided. If yes, the new blocks with actual pixel values obtained by splitting the block of $M \times N$ are introduced as candidate CU modes. This splitting process based on a recursive quad-tree is continued and the set of candidate CU modes can be obtained. By this setting, new CU modes are introduced when the width of the CU is not equal to its height. For example, when LCU is size of 64x64 and the block with actual pixel values on the frame boundary is size of 64x8, the non-square CUs of 64x8, 32x8, 16x8 are integrated into the CU candidate set according to the proposed method.

For the non-square CU coding, the prediction modes are specified for every non-square CU which is not subject to further splitting, namely, the leaf nodes of the CU hierarchical tree. The acquirement of the PU is similar to that of the method in the original HEVC, targeting inter prediction. PU modes of $2M \times 2N$, $2M \times N$, $M \times 2N$ and $M \times N$ are used for a non-square CU size of $2M \times 2N$, for instance. Correspondingly, the non-square TU would be padded into the square shape for transform. Rate-distortion optimization would be employed to determine which of the non-square CU modes is to be chosen. It should be noticed that the proposed method is well

compatible to the original CU splitting method for the non-boundary area. By this method, the coding performance would be improved by saving the bit rate for coding motion and transformation information. Fig. 4 shows the non-square CU modes used in the proposed method. It is seen that small sized CUs are reduced compared to Fig.2 and this splitting modes can be more flexible according to video content.
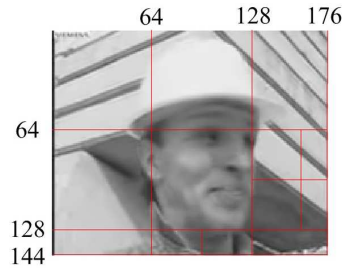

Fig. 4 CU modes for "Foreman" using the proposed method.

3. Performance evaluation

There are two coding constraint conditions for different application scenarios in HEVC: low delay and random access [4]. In this paper, the reference configuration file of "Random access high efficiency" is performed, which is used for measurement of the bit rate under different quantization parameters according to the Bjøntegaard Delta method [5].

For test sequences, JCT-VC supplies five classes of video resolution, ranging from quarter WVGA (416x240) at the low end up to areas of size 2560x1600 [6]. Only the video resolutions of size 416x240, 832x480 and 1920x1080 are used for random access evaluation, where the video resolution is not an integral multiple of the LCU size.

The proposed method is integrated into TMuC 0.9 and is compared with the original TMuC 0.9 for performance evaluation. Here, a non-square TU is padded with the mean values of rectangular residual for transformation. The experimental conditions are followed by the rules specified by JCT-VC [6], including the coding parameters.

Table 1 shows the numerical comparisons between the proposed method and the original TMuC 0.9 for video sequences with the resolution of 1920x1080. It is seen that the coding performance of the proposed method is obviously improved by an average bit-rate reduction of 1.27% compared with the reference method at the same coding quality. Furthermore, a total average performance gain of bit rate reduction by up to 0.76% can be obtained from the proposed method compared with the reference method for all test sequences of size 416x240, 832x480 and 1920x1080.

Fig. 5 shows the RD curves for the test sequence "BQSquare" with resolution of 416x240. It could be conclude that the coding efficiency of the proposed method is higher than that of the original TMuC 0.9. To summarize, the proposed method brings a considerable improvement of coding efficiency.
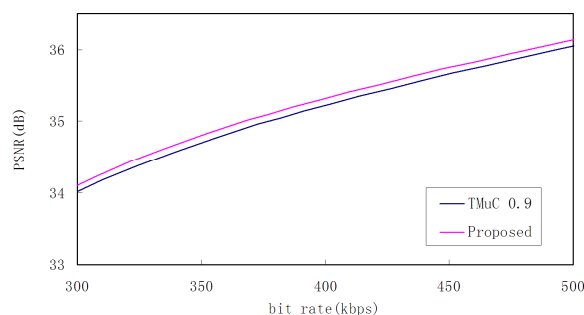

Fig. 5 The RD curves for the test sequence "BQSquare".

Table. 1 The proposed method compared with the original TMuC 0.9.

sequence	QP	TMuC 0.9		Proposed		BD bit-rate (%)
		PSNR	Bit-Rate	PSNR	Bit-Rate	
BasketballDrive	22	39.73	8933.42	39.74	8907.44	-1.00
	27	38.33	3276.12	38.34	3268.32	
	32	36.65	1580.82	36.65	1570.66	
	37	34.74	824.27	34.75	818.21	
ParkScene	22	39.95	10261.77	39.97	10197.69	-1.39
	27	37.44	4428.10	37.46	4420.09	
	32	34.81	1986.74	34.85	1975.49	
	37	32.36	890.19	32.39	885.32	
BQTerrace	22	37.96	21863.02	37.96	21750.48	-0.98
	27	35.32	5450.73	35.32	5426.74	
	32	33.62	1882.11	33.64	1874.18	
	37	31.74	830.85	31.74	826.23	
Cactus	22	38.47	13235.14	38.48	13168.59	-1.73
	27	36.81	4404.12	36.83	4382.02	
	32	34.87	2095.92	34.91	2087.58	
	37	32.68	1060.18	32.73	1052.25	
Average						-1.27

4. Conclusion

The CU splitting on the frame boundary for arbitrary resolution video is present in this paper, which adds the non-square CU modes into the CU candidate set on the frame boundary when video resolution is not an integral multiple of the LCU size. Experimental results show that an average bit rate saving of 0.76% is achieved at the same PSNR compared with the original TMuC 0.9 reference software. Meanwhile, the proposed method is well compatible to the original CU splitting method for non-boundary areas. The proposed method can be employed to improve coding efficiency on frame boundary for arbitrary resolution video coding in HEVC.

Acknowledgement

Reference

[1] T. Wiegand, J.R. Ohm, G. J. Sullivan, Woo-Jin Han, R. Joshi, T.K. Tan, K. Ugur, "Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization", Circuits and Systems for Video Technology, IEEE Transactions. No. 12, 2010, pp. 1661-1666.

[2] JCT-VC, "Draft Test Model under Consideration", JCTVC-A205, 1st JCT-VC Meeting, Dresden, April 2010.

[3] Han, W.-J., Min, J., Kim, I.-K., Alshina, E., Alshin, A., "Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools," Circuits and Systems for Video Technology, IEEE Transactions. vol. 20, 2010, pp: 1709 – 1720.

[4] Ugur, K., Andersson, K., Fuldseth, A., "Low complexity video coding and the emerging HEVC standard," Picture Coding Symposium (PCS), pp: 474, 2010.

[5] G. Bjøntegaard, "Calculation of average PSNR differences between RD-Curves," ITU-T Q.6/SG16 Doc., VCEG-M33, Austin, April 2001.

[6] JCT-VC, "Common test conditions and software reference configurations", JCTVC-C500, 3rd JCT-VC Meeting, Guangzhou, Oct, 2010.