Available online at www.sciencedirect.com

Journal of Applied Research and Technology

Journal of Applied Research and Technology 13 (2015) 205-219

www.jart.ccadet.unam.mx

Original

Feature-based fast coding unit partition algorithm for high efficiency video coding

Yih-Chuan Lin*, Jian-Cheng Lai

Department of Computer Science and Information Engineering, National Formosa University, Yunlin 63201, Taiwan

Received 19 April 2014; accepted 18 August 2014

Abstract

High Efficiency Video Coding (HEVC), which is the newest video coding standard, has been developed for the efficient compression of ultra high definition videos. One of the important features in HEVC is the adoption of a quad-tree based video coding structure, in which each incoming frame is represented as a set of non-overlapped coding tree blocks (CTB) by variable-block sized prediction and coding process. To do this, each CTB needs to be recursively partitioned into coding unit (CU), predict unit (PU) and transform unit (TU) during the coding process, leading to a huge computational load in the coding of each video frame. This paper proposes to extract visual features in a CTB and uses them to simplify the coding procedure by reducing the depth of quad-tree partition for each CTB in HEVC intra coding mode. A measure for the edge strength in a CTB, which is defined with simple Sobel edge detection, is used to constrain the possible maximum depth of quad-tree partition of the CTB. With the constrained partition depth, the proposed method can reduce a lot of encoding time. Experimental results by HM10.1 show that the average time-savings is about 13.4% under the increase of encoded BD-Rate by only 0.02%, which is a less performance degradation in comparison to other similar methods.

Keywords: HEVC; CTB; Coding unit; Sobel edge; Feature-based approach; Fast algorithm; Quad-tree partition

1. Introduction

With rapidly increased demand for higher visual quality in consumer devices, the video coding standard H.264/AVC has been becoming insufficient in terms of rate-distortion coding performance. The new HEVC (or called H.265) standard was established by ITU-T VCEG and ISO/IEC international organizations to meet the newest visual coding requirements for ultra high definition videos (Sullivan et al., 2012). HEVC employs a quad-tree based coding block structure to increase the rate-distortion performance. The types of quad-tree block defined in HEVC include CU, PU and TU blocks. The CU in HEVC is the basic coding block, similar to the macroblock in H.264/AVC, except that CU could be split further into PU and TU blocks in HEVC. The role of PU is to help get a good prediction of image blocks based on a predefined set of 35 prediction modes. And TU is a partition of prediction residue for a CU in order for obtaining better DCT/DST transform performance. Table 1 shows that the possible block sizes of CU, PU, TU and the corresponding distinct quad-tree depth, respectively. The symbol "V" represents the size supports for the block type and "X" means not supported for the block type.

Figure 1 shows an example of quad-tree partition for a block of 64 × 64 pixels, in which it should execute 341 times process of PU, TU and encoding for the incoming block. A lot of encoding time is required for this coding process, and then it limits the possibility of HEVC in real-time applications. There are some fast algorithms proposed to speedup the HEVC encoder. Lee et al. (2012) introduces the co-located CU in previous frame to predict the quad-tree depth of current encoded CU block. Cheng et al. (2012) and Shen et al. (2013) apply neighbouring CUs and parent CU information of quad-tree partition to estimate the possible split of current block and prediction mode of intra coding, which can provide up to 40% and 20% time-savings, respectively. Shen et al. (2013) and Zhang et al. (2013) present the neighbour reference method to decide the CU depth level in the inter coding mode. These methods can easily and quickly achieve the time-savings and the compression ratio. However, in some cases, the estimation is not always accurate.

Table 1
Block sizes for different block types.

Block size/depth	CU	PU	TU
64 × 64/0	V	V	X
32 × 32/1	V	V	V
16 × 16/2	V	V	V
8 × 8/3	V	V	V
4 × 4/4	X	V	V

*Corresponding author.
E-mail address: lyc@nfu.edu.tw (Y.-C. Lin).

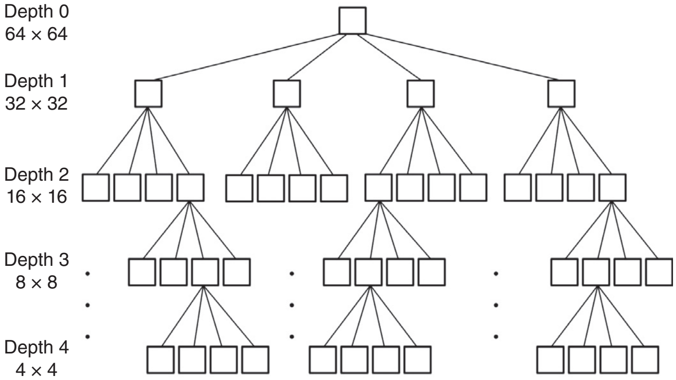

Fig. 1. Quad-tree coding partition example.

For example, in frames with fast moving objects or blocks located at frame boundary (Fig. 2, green rectangles) do not have reliable adjacent reference blocks for estimation. This means that there is no significant time-savings on these blocks. In order to solve the above-mentioned problem, the non-reference adjacent block information method was proposed. Some methods adopted RDCost (rate-distortion cost) to early terminate the split of CU (Kim et al., 2013; Zhang & Ma, 2013). They used RDCost and threshold values to derive the most probable split of block. The average time-savings is up to 20% and 60%, respectively. Zhao et al. (2012) employs QP (quantization parameter) value to decide the possible block depth level. Choi and Jang (2012) utilized the nonzero DCT coefficients in TU to terminate the determination of TU partition. Other methods were proposed to reduce PU partition options and prediction modes. For example, Yan et al. (2012) proposed to utilize RMD to reduced RDO calculation. Da Silva et al. (2012) and Jiang et al. (2012) also present the filter-based algorithm to find the best intra prediction mode. These methods are accompanied with different thresholds to terminate the calculation. But they do not require the neighboring block information.

As for the determination of threshold values, there are two categories of method: adaptive and non-adaptive. The adaptive threshold method is able to vary threshold values during the coding process in accordance with the variation of video content. It has a higher practicability to account for the variation in videos. The non-adaptive threshold method uses fixed threshold values for a single video coding process. The thresholds are usually determined from extensive experiments and analysis. This approach would has a best compression and time-savings in particular sequence. This paper focuses on the non-adaptive threshold method. The thresolds are varied with QP values setting before encoding each video.

Fig. 2. Boundary partition blocks.

This paper proposes a fast feature-based algorithm for CTB depth level estimation. The concept of the proposed method is based on our previous work (Lin & Lai, 2014) that uses image features to early terminate the partition process of CUs for speeding up the HEVC encoder. For image processing, image features can really provide effective tools for estimating the visual content of pixels in an image patch (Yasmin et al., 2013, 2014). In this paper, an in-depth elaboration of how feature-based CU partition employs image features to speedup CU partition and an extensive experiment are provided to evaluate the fast feature-based CU partition method. In the subsequent parts, Section 2 introduces the HEVC intra coding process. Then, Section 3 presents the proposed fast algorithm method. Section 4 shows the experimental results and makes comparison to another method. Finally, Section 5 concludes the paper with remarks and future works.

2. HEVC intra coding process

Figure 3 shows a block diagram of the HEVC encoder architecture. Firstly, it divides the incoming frame into a set of CTB blocks and encodes each CTB at a time in the subsequent functional blocks. The main function of intra coding lies in the block intra-prediction to remove the redundancy among spatial neighboring pixels in a single video frame. The intra prediction of HEVC employs different angular prediction modes to make smaller prediction residual signal for the block. On the other hand, the inter coding implements the motion estimation and motion compensation to generate a better prediction of blocks with referring to temporally neighboring pixels across neighboring frames. This approach is similar to the inter coding part of H.264/AVC. Finally, the prediction result and the associated prediction parameters are transfered to entropy coding functional block to complete the encoding of the frame.

In this section, the intra coding process in HEVC is mainly described. As shown in Figure 4, the incoming frame is firstly split into a series of 64 × 64-blocks, each of which is called the largest CU (LCU) and would be processed individually in the encoder. In intra mode prediction, HEVC provides 35 angular prediction modes (Fig. 5, bottom-left) for predicting current CU by using its neighboring pixels. The HEVC standard defines the rough mode decision (RMD) process (Piao et al., 2010) to reduce the 35 prediction modes into *N* candidate modes before getting the best prediction. The possible values of *N* are defined by Table 2. For the larger CU, such as larger than 16 × 16 size, *N* is set to a smaller value of 3; otherwise, *N* is 8. The rate-distortion optimization (RDO) process performs to select a best mode from the *N* candidate modes.

To obtain a possibly better RD performance, the LCU is tried to be partitioned into four smaller blocks (Fig. 5, upper-