Entropy **2013**, *15*, 2277-2287; doi:10.3390/e15062277

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

Entropy-Based Fast Largest Coding Unit Partition Algorithm in High-Efficiency Video Coding

Mengmeng Zhang [1,*], Jianfeng Qu [1] and Huihui Bai [2,*]

[1] College of Information Engineering, North China University of Technology, Beijing 100144, China; E-Mail: qjf_sharpsword@126.com

[2] Institute of Information Science, Beijing Jiaotong University, Beijing 100144, China

* Authors to whom correspondence should be addressed; E-Mails: zmm@ncut.edu.cn (M.M.Z.); luckybhh@gmail.com (H.H.B.); Tel.: +86-10-8880-3055; Fax: +86-10-6887-5846.

Received: 3 April 2013; in revised form: 22 April 2013 / Accepted: 30 May 2013 / Published: 6 June 2013

Abstract: High-efficiency video coding (HEVC) is a new video coding standard being developed by the Joint Collaborative Team on Video Coding. HEVC adopted numerous new tools, such as more flexible data structure representations, which include the coding unit (CU), prediction unit, and transform unit. In the partitioning of the largest coding unit (LCU) into CUs, rate distortion optimization (RDO) is applied. However, the computation complexity of RDO is too high for real-time application scenarios. Based on studies on the relationship between CUs and their entropy, this paper proposes a fast algorithm based on entropy to partition LCU as a substitute for RDO in HEVC. Experimental results show that the proposed entropy-based LCU partition algorithm can reduce coding time by 62.3% on average, with an acceptable loss of 3.82% using Bjøntegaard delta rate.

Keywords: HEVC; entropy; coding unit; information content

1. Introduction

As the next generation of video coding standards, high-efficiency video coding (HEVC) [1] aims to reduce bit rate in half with the same reconstructed video quality as H.264/AVC, which is the latest-generation video coding. Many useful tools are adopted in HEVC, such as Sample Adaptive Offset, Motion Vector Merging, Merge Skip and Residual Quadtree Transform [2].

HEVC provides a larger coding unit (CU), which is fixed 16×16 size in H.264, and a more flexible quadtree structure. The CU sizes of HEVC are 128 × 128, 64 × 64, 32 × 32, 16 × 16, 8 × 8, and 4 × 4. One Largest Coding Unit (LCU) can be split into four equal-sized CUs, and one CU can be encoded or split into four equal-sized CUs [3]. This split only ends when the CU reaches the smallest CU. To find the optimized combination of CUs, the encoder has to fully search all possible CUs. Figure 1 shows an LCU partitioned into CUs. Whether or not a CU whose size is larger than smallest CU is encoded or split into four equal-sized CUs is decided by using a rate distortion optimization (RDO). This ergodic process searches and encodes all possible CUs to choose the CUs with the smallest rate distortion (RD) cost. Using a more flexible quadtree structure results in a more efficient encoder [5] and can bring coding gain by effectively adapting the diversity of picture content. According to [6], a new coding tree structure with a 64 × 64-sized LCU can bring nearly 12% bitrate reduction on the average compared with a 16 × 16-sized LCU.

Figure 1. Quadtree structure of HEVC.

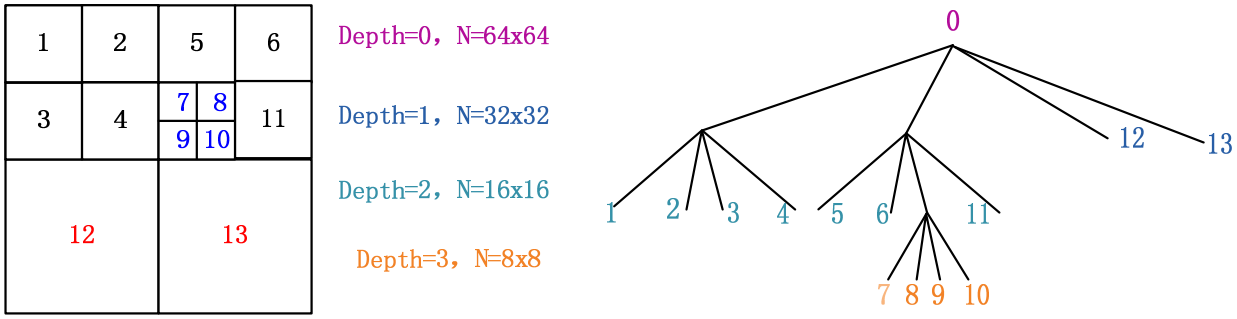


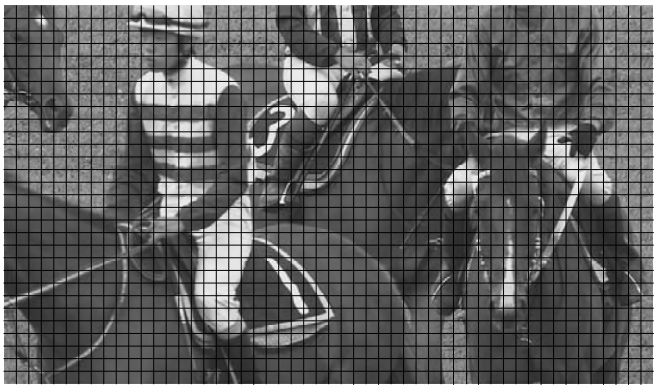
Figure 2 shows an image partitioned by H.264 and HEVC. In Figure 2b, the red lines represents the edge of LCUs (64 × 64) and the larger CU results in a more focused encoder [7]. The areas with lesser information content are partitioned into larger CUs. By contrast, the areas with more information content are partitioned into smaller CUs.

Although a larger CU can bring significant bitrate reduction, the HEVC encoder has to search for all possible CUs to obtain the optimized CUs, resulting in an extremely large computation complexity [8]. To find the optimal CUs, the computation burden is equivalent to encoding an LCU four times because the encoder has to encode the CUs with sizes 64 × 64, 32 × 32, 16 × 16, and 8 × 8. Given that an encoder creates the optimal partition plan only once, 75% of the computation burden is therefore wasted. The large burden computation is not appropriate for many applications of video coding, such as real-time application scenario. We thus proposed a new algorithm to avoid the large computational redundancy in encoding.

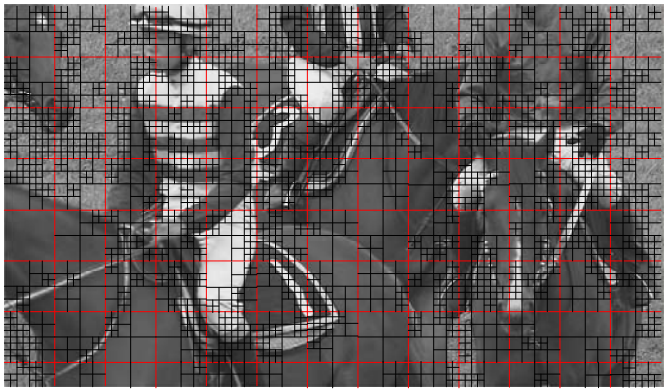
Some related proposals on complexity reduction for intra coding in HEVC. Cho [9] proposed a fast splitting and pruning method which performed in two complementary steps: (i) early CU split decision; and (ii) early CU pruning decision. Piao *et al.* [10] presented a rough mode decision (RMD) method to prescreen the most possible candidate modes for the intra prediction coding of HEVC by computing low-complexity RD cost. Shen [11] proposed a CU size decision algorithm, which collects relevant and computational-friendly features to assist decisions on CU splitting. These related works can only

got a 50% coding time reduction at most, and none of them take the information content of CU into account of LCU partition.

Figure 2. Example of the CU partitioning of HEVC: (**a**) Partitioned by H.264 and (**b**) Partitioned by HEVC.



(a)



(b)

In this paper, we propose an entropy-based fast CU-sized decision algorithm to replace the RDO used in the quadtree structure. This paper is organized as follows: Section 2 briefly introduces the principle of proposed algorithm. In Section 3, we elaborate on the entropy-based fast CU-sized decision algorithm. Experimental results are shown in Section 4, and Section 5 concludes our study.

2. Principle of the Proposed Algorithm

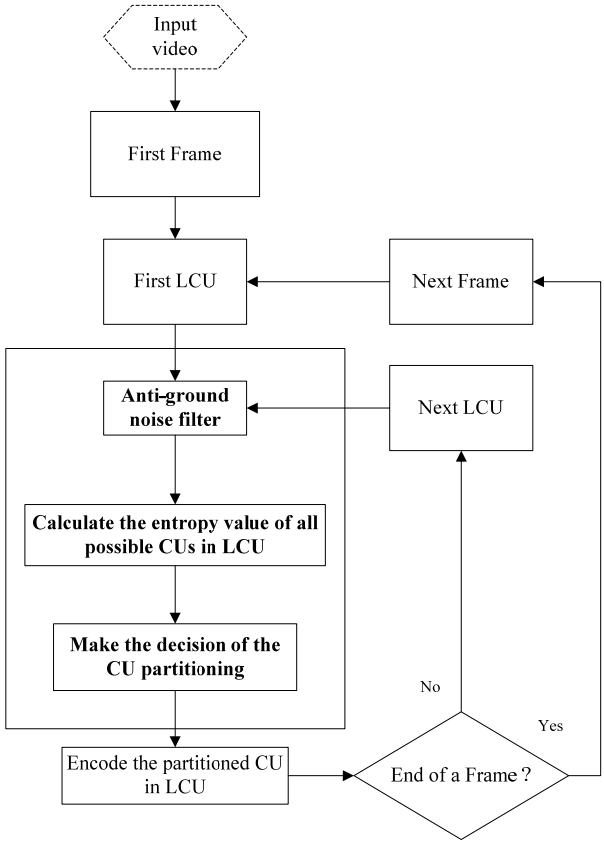
This study shows that the CUs partitioned by RDO process closely relate to the information content of each CU. Figure 2b shows that the partition of LCUs is related to the information content. Given that Shannon entropy is the average unpredictability in a random variable, which is equivalent to its information content, this paper proposed a Shannon entropy [12] technique to replace the RDO in LCU partition.

3. Proposed CU-Sized Decision Algorithm

The proposed algorithm is introduced in this section. Figure 3 shows the flowchart of our proposed algorithm.

As introduced in the previous section, the key point of the proposed algorithm is to find the relationship between the selected CUs and the entropy of these CUs. Based on this point, the CUs partitioned by using the proposed algorithm can have maximum similarity to the optimized CUs, which is the aim of this study.

Figure 3. Flowchart of the proposed algorithm.



3.1. Entropy of Each CU

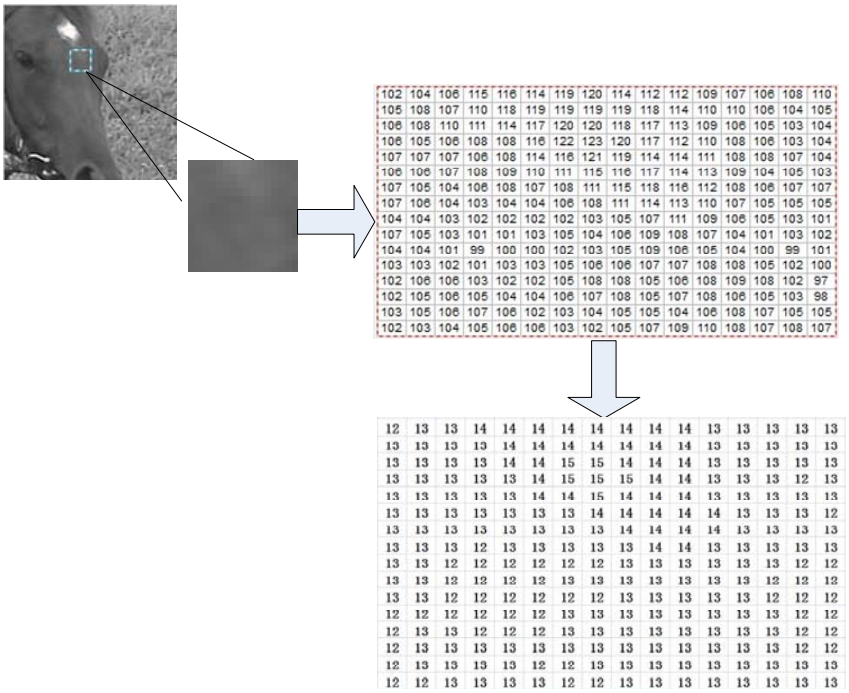
This section shows the calculation of the CU-sized decision algorithm. The equation for the entropy is expressed as follows:

$$H(x) = -\sum_{i=0}^{i=j} p_i \log_2 p_i \quad (1)$$

In this equation, H(x) is the entropy, p presents the probability of the factor i, and j is the number of factors. To obtain the information content of CUs, we calculated the entropy of all the possible CUs in an LCU. However, before the calculation, the background noise of the LCU was first dislodged. The background noise is the pixel value difference that should not exist among neighbor pixels. Figure 4 shows that the area is very smooth and the pixel values seem to be the same. However, because of the background noise, the pixel values slightly differed from one another, which resulted in an inaccurate description of the information content using entropy. To dislodge the background noise without high computation complexity, we used an anti-ground noise filter. We adopted 8 as the stepper to quantize

all the 256 pixel values. Up to 32 (0–31) pixel values were left, which provided a good condition for the subsequent work.

Figure 4. Example of the anti-ground noise filter.



We then calculated the entropy of all possible CUs in the LCU. A total of 85 possible CUs were available in the LCU, including one 64 × 64-sized CU, four 32 × 32-sized CUs, sixteen 16 × 16-sized CUs, and sixty-four 8 × 8-sized CUs . We then counted the possibility of the appearance of each pixel in a CU. This possibility was used in the calculation of the entropy. The equation of the possibility count is expressed as:

$$p_i = \frac{n_i}{N} \ (i = 0,1, \dots \dots 31) \tag{2}$$

where N is the number of pixels in a CU and n_i is the number of the pixels whose values are i.

We then calculated the entropy of each CU using the following equation:

$$H(x) = -\sum_{i=0}^{i=j} p_i \log_2 p_i \ \ (j = 31) \tag{3}$$

A total of 85 values for the entropy were calculated, and the results were used as the base for the CU partition.

3.2. Threshold and Judgment of the Proposed Algorithm

Based on the theory in Section 2 and the entropy of all the possible CUs, we conducted several experiments to find the relation between the entropy of each CU and the CUs which were partitioned by RDO. To achieve maximum similarity between the proposed CUs and the optimized CUs, we established some rules and thresholds. Through our search, we found these principles to partition LCU:

a. If the entropy of the CU is extremely small, the CU is likely to terminate its partition.
b. If the entropy of the CU is extremely large, the CU is likely to be partitioned.
c. The CUs whose entropy is approximately the average of all the possible CUs typically appear in the final partitioning map.

Based on these rules, we searched for several thresholds. Taking every video sequence into the consideration, we count the relationship between the optimal CUs which is partitioned by HEVC and the corresponding entropy values. After that we count the entropy value of the CU which is split in HEVC. As we got two kinds of entropy values, we sort these entropy values separately. Then we use formulas (4)–(6) to calculate the threshold for CU partitioning:

$$T_a = \min(E_{optimal60\%}, E_{split10\%}). \quad (4)$$

$$T_b = \max(E_{optimal10\%}, E_{split60\%}). \quad (5)$$

$$T_c = \min(|E_{Average} - E_{optimal90\%}|, |E_{Average} - E_{split90\%}|). \quad (6)$$

T_a, T_b, T_c, represent the thresholds for principle a, b, and c. $E_{optimalxx\%}$ presents the top xx% entropy value in the sort ascending of optimal CU entropy values. $E_{splitxx\%}$ presents the bottom xx% entropy value in the sort ascending of split CU entropy values. For example, if there are 24 optimal CUs, 36 split CUs in a LCU, $E_{optimal60\%}$ means the 14th entropy value in the sort ascending of optimal CU entropy value, $E_{split60\%}$ means the 21st entropy value in the sort ascending of optimal CU entropy value. $E_{Average}$ presents the average of all the entropy values in a LCU. The percentage values, 60%, 10%, 90%, are got with the consideration of the loss of BD-rate.

During our research, we found thresholds , which can provide the maximum similarity between the proposed CUs and optimized CUs are almost the same in most video sequences. So, with the consideration of every video sequence, we get the best thresholds:

a. CUs whose entropy is smaller than 1.2 will not be partitioned.
b. CUs whose entropy exceed 3.5 will be partitioned.
c. CUs whose entropy is 0.15 bigger or 0.15 smaller than the average entropy will not be partitioned.

We can distinguish whether a CU is split or not by determining the thresholds. Thus, we can partition an LCU right after we obtain the entropy values. Figures 5(a,b) show the difference between the proposed CUs and the optimized CUs. The parts with red lines in Figure 5(a) are the parts that did not match with the optimized CUs. Figures 5(a,b), as an example, show the similarity between proposed CUs and optimal CUs. Table 1 shows that the proposed algorithm obtained nearly 70% similarity to the partition of the RDO on average in HEVC. The similarity is calculated by formula (7):

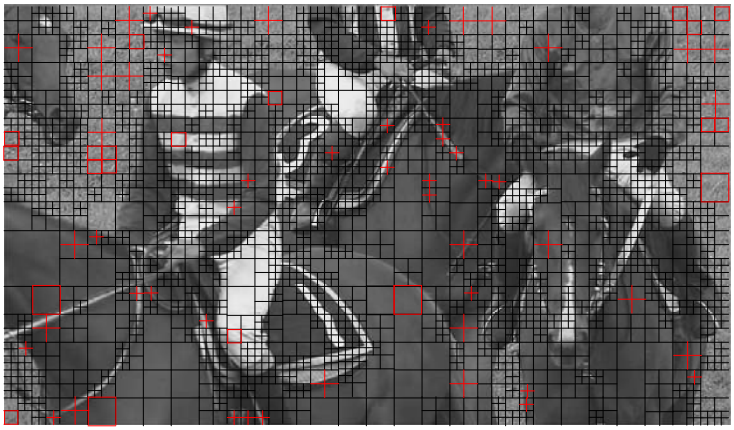
$$S = \frac{n_{match}}{N_{CU}} \quad (7)$$

N_{CU} represents the number of CU in a LCU, n_{match} represents the number of CU which matches the optimal CU.

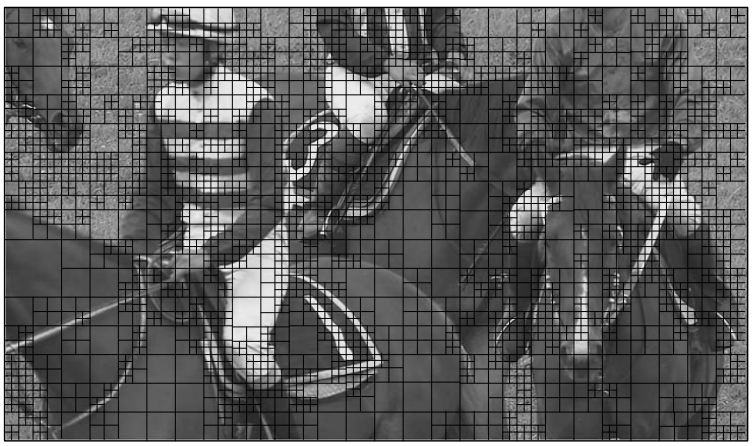
Table 1. The similarity between proposed CUs and optimal CUs.

Seq.Name	Similarity(%)
Traffic_2560×1600_30_crop	69.3
ParkScene_1920×1080_24	70.4
BQTerrace_1920×1080_60	72.2
BQMall_832×480_60	65.2
RaceHorses_832×480_30	67.6
BQSquare_416×240_60	71.6
BasketballPass_416×240_50	70.2
BlowingBubbles_416×240_50	65.1
RaceHorses_416×240_30	70.6
Average	69.1

Figure 5. (**a**) CU presentation of the sequence RaceHorsesC optimized by HM10.0 using the proposed algorithm. (**b**) CU presentation of the sequence RaceHorsesC optimized by HM10.0 with RDO.



(a)



(b)

4. Experimental Results

Up to 300 frames of each sequence were coded to test the performance of the proposed algorithm, and the test condition is“All Intra–Main” (AI-Main) [13]. QP values are set to 22, 27, 32, 37.

A computer with a 2.8 GHz core was used in this experiment. To fully determine the performance of the proposed algorithm, we used HM10.0 with 16 × 16-sized LCU for the comparison.

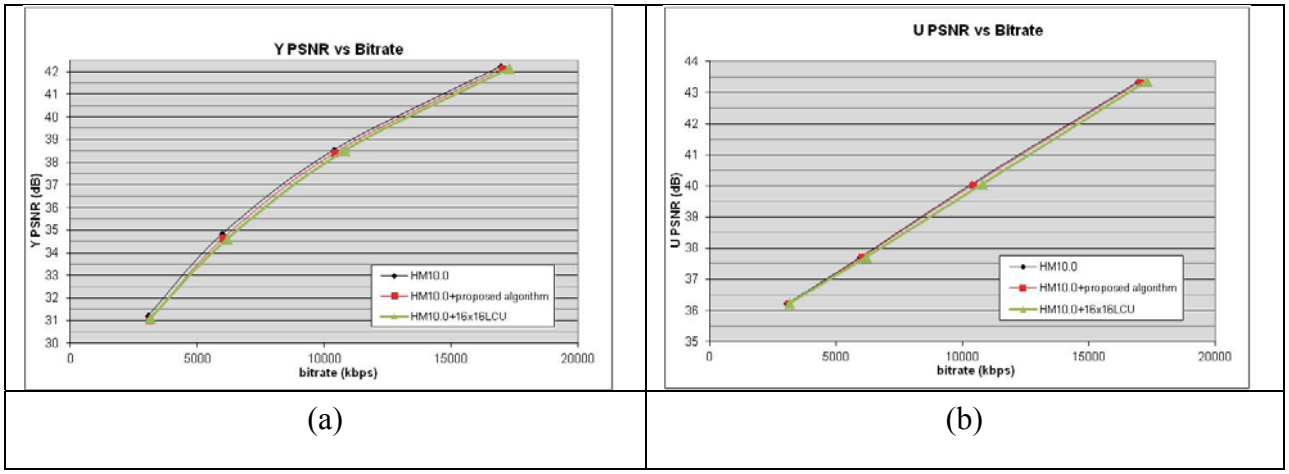
We used Equation 7 to measure the reduced coding time:

$$\Delta T = \frac{T_{HM10.0} - T_{Proposed}}{T_{HM10.0}} \quad (8)$$

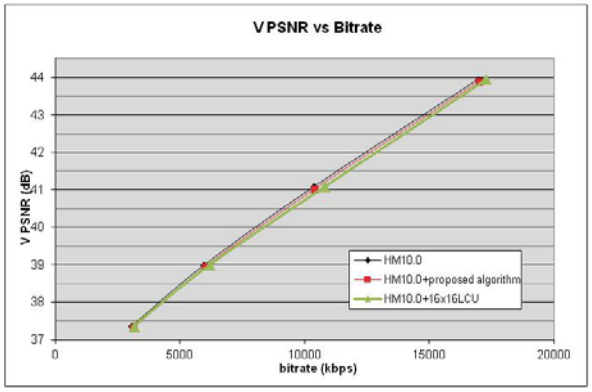
$T_{HM10.0}$ is the coding time of HM10.0 with RDO, $T_{Proposed}$ is the coding time of HM10.0 with the proposed algorithm, and ΔT stands for the time reduction.

Table 2 shows that on average, the coding time of HM10.0 resulted in a 62.0% reduction. The Bjøntegaard delta (BD) rate [13] exhibited a 3.68% loss. A 0.10% decrease at PSNR was also observed. Figure 6 shows the curves of HM10.0 with the proposed algorithm, HM10.0 and HM10.0 with 16 × 16-sized LCU. Figure 6 shows that the proposed curve was extremely near the curve of HM10.0 and was better than the curve of HM10.0 with small LCU (16 × 16).

Figure 6. Experimental results of "RaceHorsesC" (832 × 480) under different QP settings (22,27,32,37). (**a**) Y PSNR *versus* Bitrate; (**b**) U PSNR *versus* Bitrate; (**c**) V PSNR *versus* Bitrate.



(a) (b)



(c)

Table 2. Results for the proposed algorithm.

	Proposed techniques							**HEVC with 16x16 size LCU**						
Seq. Name	Y-BD-rate (%)	Y-PSNR (%)	U-BD-rate (%)	U-PSNR (%)	V-BD-Rate (%)	V-PSNR (%)	△T (%)	Y-BD-rate (%)	Y-PSNR (%)	U-BD-rate(%)	U-PSNR (%)	V-BD-rate (%)	V-PSNR (%)	△T (%)
BasketballDrive	3.3	−0.13	2.3	−0.07	1.6	−0.05	65.2	9.7	−0.09	20.0	−0.03	18.9	−0.05	67.6
BQTerrace	3.2	−0.07	1.7	−0.03	1.9	−0.03	67.3	3.3	−0.06	6.6	−0.05	14.2	−0.03	68.5
ParkScene	3.8	−0.12	1.9	−0.06	1.3	−0.07	64.0	3.9	−0.09	3.7	−0.03	5.6	−0.02	67.1
Group Average	3.43	−0.11	1.97	−0.053	1.2	−0.05	65.9	5.63	−0.08	10.1	−0.04	12.9	−0.033	67.7
BasketballDrill	3.6	−0.07	1.3	−0.04	1.7	−0.05	61.1	3.2	−0.11	2.4	−0.08	1.8	−0.08	68.7
BQMall	3.7	−0.10	2.1	−0.07	2.0	−0.06	61.2	3.9	−0.13	5.9	−0.03	7.1	−0.03	66.2
PartyScene	4.1	−0.11	2.7	−0.05	1.8	−0.03	59.6	4.5	−0.09	1.1	−0.05	1.6	−0.05	71.2
RaceHorsesC	4.0	−0.10	2.5	−0.06	1.9	−0.05	63.2	4.3	−0.11	4.9	−0.04	5.4	−0.03	63.3
Group Average	3.85	−0.09	2.15	−0.055	1.85	−0.0475	62.4	3.98	−0.11	3.6	−0.05	4.0	−0.048	67.35
BasketballPass	3.4	−0.12	2.3	−0.10	2.1	−0.09	57.9	3.8	−0.10	6.4	−0.04	5.1	−0.04	70.3
BlowingBubbles	4.4	−0.13	2.2	−0.11	1.7	−0.08	59.1	4.3	−0.11	4.1	−0.01	4.2	−0.01	64.5
BQSquare	3.8	−0.08	3.0	0.05	2.7	−0.04	56.2	3.9	−0.09	0.5	−0.05	0.6	−0.08	67.3
RaceHorses	3.5	−0.07	2.1	0.04	2.0	−0.04	59.6	3.7	−0.08	2.2	−0.02	2.1	−0.05	68.9
Group Average	3.77	−0.10	2.4	0.075	2.13	−0.0625	57.8	3.93	−0.10	3.3	−0.03	3.0	−0.045	67.75
Total Average	3.68	−0.10	2.17	−0.061	1.73	−0.053	62.0	4.51	−0.10	5.67	−0.04	6.63	−0.042	67.6

5. Conclusions

In this paper, we propose a new technology to partition LCU in HEVC. The proposed algorithm aims to highly reduce the computation complexity with an acceptable loss of BD rate. Based on the results shown, the proposed algorithm significantly reduced the coding time with an acceptable decrease in quality, therefore, entropy-based fast LCU partition is a topic worthy of further investigation.

Acknowledgements

This work is supported by the Natural National Science Foundation of China (No. 61103113 and No.61272051), Jiangsu Provincial Natural Science Foundation (BK2011455) and Beijing Municipal Education Commission General Program (KM201310009004).

Conflict of Interest

The authors declare no conflict of interest.

References

1. Han, W.-J. Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools. *IEEE T. Circ. Syst. Vid.* **2010**, *12*, 1709–1720.
2. I1-Koo Kim. High efficiency video coding (HEVC) test model 10 (HM10) encoder description. In Proceedings of the 12th JCT-VC Meeting, Geneva, Switzerland, 14-23 January 2013; pp. 27–28.
3. Bross, B. High efficiency video coding (HEVC) text specification draft 7. In proceedings of the 5th JCT-VC Meeting, Geneva, Switzerland, 27 April–7 May 2012; pp. 34–39.
4. Cassa, M.B. Fast rate distortion optimization for the emerging HEVC standard. In proceedings of Picture Coding Symposium (PCS), Kraków, Poland, 7–9 May 2012; pp.493–496.
5. Han, W.-J. Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools. *IEEE T. Circ. Syst. Vid.* **2010**, *12*, 1709–1720.
6. Kim, M.K.J. JCTVC-C067 TE9: Report on large block structure testing. In proceedings of Third meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Guangzhou, China, 7–15 October 2010; pp. 2–4.
7. Kim, I. Block partitioning structure in the HEVC standard. *IEEE T. Circ. Syst. Vid.* **2012**, 12, 1697–1706.
8. Sullivan, G.J.; Wiegand, T. Rate-distortion optimization for video compression. *IEEE Signal Proc. Mag.* **1998**, *15*, 74–90.
9. Cho, S. Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. *IEEE T. Circ. Syst. Vid.* **2013**, *2*, 1–10.
10. Piao, Y.; Min, J.H.; Chen, J. Encoder improvement of unified intra prediction. In proceedings of JCT-VC 3rd Meeting, Guangzhou, China, 7–15 October 2010; pp. 1–3.

11. Zhao, L.; Zhang, L.; Ma, S.; Zhao, D. Fast mode decision algorithm for intra prediction in HEVC, In Visual Communications and Image Processing (VCIP), Proceedings of the 2011 IEEE, Tainan, Taiwan, 6-9 November 2011; pp. 1–4.
12. Shunsuke, I. *Information Theory for Continuous Systems*; World Scientific: Singapore, Singapore, 1993; Volume 1, pp. 2–3.
13. Bossen, F. JCT-VC-L1100: Common test conditions and software reference configurations. In proceedings of JCT-VC 12th Meeting, Geneva, Switherland, 14–23 January 2013; pp. 2–4.

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).