An Efficient CU Partition Algorithm for VVC Intra Coding

Dongqi Zhang and Qiang Li*

Department Communications and Information Engineering, Chongqing University of Posts and Communications, China

*Corresponding author email: liqiang@cqupt.edu.cn

Abstract. VVC currently under development is the latest video coding standard. VVC adopts many new coding technology or improve the original technology to enhance the coding efficiency on the basis of HEVC, at the cost of significantly increased computational complexity, which limits the industrialization and application of the VVC. To address this issue, an efficient CU partition method is presented. This method can effectively reduce computational complexity while preserving encoding efficiency. The proposed method utilizes the texture information and residual coefficient distribution of the CU to skip the low-probability CU partition types in advance. Experimental results illustrate that under the common test conditions, our proposed fast algorithm achieves 52.3% of the coding time saving while the BD-BR only increases by 1.31%, and the objective quality of the video remains basically unchanged.

Keywords: VVC; CU split; Intra prediction.

1. Introduction
Due to the widespread popularity of high-resolution, high frame rate video, the volume of data stored and transmitted for video has increased significantly. The High Efficiency Video Coding (HEVC) standard [1] has insufficient compression capacity to cope with the rapidly increasing data volume. To address this issue, Joint Video Exploration Team (JVET) start to develop the next-generation of video coding standard in 2015. In 2018, the committee had published Call for Proposals (CfP) [2] and named the next-generation coding standard as Versatile Video Coding(VVC), aiming to enhance the coding efficiency by 50% on the basis of HEVC. In the working draft of VVC [3], adopts many new coding technology or improve the original technology to enhance the coding efficiency. For instance, the quadtree with multi-type tree (QTMT) block partition structure [4], Multiple transform selection (MTS) [5], adaptive multiple core transform and Position Dependent intra Prediction Combination (PDPC) [6], etc. It is reported that these techniques enhance the coding efficiency in VVC, but the cost of coding computational complexity has increased significantly.

The core of video coding is to obtain the optimal coding block partition structure, because a good block division can bring a great increase in coding efficiency. The VVC adopts the QTMT to partition the picture, which makes the structure of CU more flexible and improves coding efficiency. During the encoding process, horizontal and vertical binary tree(BT), ternary tree(TT) can be sequentially performed on the leaf nodes of the quadtree. In the processes of CTU partition, if the multi-type tree is used to partition the CU, quadtree will no longer be used in the sub-CU partitioning procedure. In addition, the sizes of the coding block would be limited, the size of CU range from $4 \times 8$ at minimum to $128 \times 128$ at maximum. Leaf node trees are called coding units. In VVC, they are also used for transformation and prediction. Figure.1 is an instance of the CTU partition structure.
Although the application of the QTMT partition structure enhances the performance of the VVC, but at the cost of calculation complexity were significantly increased also. When the CTU is divided, the CTU is first partitioned into 4 square CU which size is 64×64, and then all CU depth need to be traversed to find the optimal partition structure. Although this method improves coding efficiency, it also takes a lot of time. If we can skip the partitioning types that save a lot of time but have a low probability during the partition process, it will save a lot of time and will basically not affect the coding efficiency of the video. Although emerging standards also follow the hybrid video coding framework based on predictive plus transform, the CU early termination algorithm for HEVC may not perform well in VVC. Therefore, it is necessary to research a general and effective algorithm to obtain the optimal CTU partition.

There are many researchers who are devoted to improving the encoding speed while maintain the encoding efficiency. Some methods (such as [7] and [8]) do not need to traverse all the partitioning and pruning processes, so the computational complexity is reduced. But if we use the generated encoding information and the texture information of CU to skip certain partitions in the encoding process, the coding time would be effectively saved. The first section of the article is the background and introduction of VVC. The second section introduces the fast CU partition method from two aspects. The third section analysis the experimental results under common test conditions (CTC) [9], and then the fourth section summarizes the paper.

2. Proposed Algorithm
In this section, we proposed an efficient CU partition algorithm. The algorithm will skip low-probability partition types, and realize rapid CU partition structure decision in the coding process.

2.1. Early Skip for Multi-type Partition Based on CU Texture Complexity
In VVC, pictures are coded in units of blocks. When encoding a picture, for a CU in homogeneous region, a larger CU size tends to be chosen, and for a CU with rich texture, a smaller CU is chosen [10]. For a CU with simple or consistent texture, the pixel value in CU can be accurately predicted by one kind of intra prediction mode.
In natural pictures, CU with simple texture have similar in the pixel value. Therefore, we can use the absolute difference between the statistical data and its average value to detect the similarity of CU. We can defines the similarity of the CU as
Among them, $\mu$ is the mathematical expectation of the sample, $N$ represents the number of samples, and $X$ represents the sample value. The average pixel value of CU is first calculated, then the mean of the absolute difference between each pixel in the CU and the average pixel is calculated. The smaller $D(x)$, the simpler the texture of CU. If the average value of the absolute difference between sample and the mean is in a certain area, the texture of current CU can be considered simple. Therefore, the value of $D(x)$ should satisfy

$$\frac{D(x)}{\mu} \leq T$$

In order to obtain the most suitable threshold, the BasketballDrive, BasketballDrill and PartyScene sequences was selected for testing, and the result is shown in figure 2. Among them, the abscissa represents the threshold, and the ordinate represents the coding loss.

It can be seen from figure 2 that the threshold increased with the coding efficiency loss increased. When the threshold is 0.01, the video coding loss can be ignored. When the threshold is less than 0.1, the coding efficiency loss slowly increases. If the threshold is more than 0.1, the coding efficiency loss increases sharply. If the threshold is more than 0.2, the coding efficiency loss is high and basically remains unchanged, indicating that the CU cannot be accurately predicted. In order to ensure the coding efficiency, the algorithm uses 0.1 as the threshold. If the given threshold is met, it can be considered that the current CU texture is simple and the CU partition can be terminated early. However, for maintain the robustness and versatility, if the CU has simple texture, the quadtree partition types will be executed and multi-type partition types will be skipped.

Figure 2. Threshold and corresponding coding loss.

2.2. Early Skip for Partial Partition Based on CU Texture Complexity

Although CU with simple textures are partitioned fast, region with rich textures account for a high proportion of the picture, and computational complexity in the encoding process is still high. Therefore, we need an algorithm to fast splitting for CU with complex textures. For CU with complex textures, if we can determine the current CU partition types before encoding, we can reduce unnecessary partition types attempts, thereby reducing the computational complexity of encoding.
If current prediction mode is optimal, the residual coefficients of CU compliance with uniformly distributed [11], [12]. Based on this, the residual coefficients from the current CU can be symmetrically divided into two different partitions, such as a left partition and a right partition, as showed in figure 3. Then we will introduce a statistical test method to detect the similar of two different partitions. In the same way, the current CU would also be vertically split into two different partitions. As showed in figure 4, and the similarity between the two partition will be detected. If the distribution similarity of the residual coefficients in top partition and bottom partition is higher, the horizontal splitting types will be skipped. On the contrary, if the similarity of the residual coefficients in the left partition and right partition is higher, the vertical splitting types will be skipped. Since the residual coefficient compliance with to the normal distribution, it is assumed that the residual distribution of different partitions in the current CU is

$$X_1 \sim N\left(\mu_1, \sigma_1^2\right), X_2 \sim N\left(\mu_2, \sigma_2^2\right)$$

Where $\mu_1$ and $\mu_2$ are respectively the mathematical expectation of two parts, and the $\sigma_1$ and $\sigma_2$ are the standard deviation of two partitions. $X_1$ and $X_2$ express a normal distribution of different partitions, and $N$ is the number of sample. For example, in the left-right partitions, where $X_1$ is the normal distribution of the left partition, $X_2$ is the normal distribution of the right partition.

To check the similarity of residual coefficients in two partition of current CU, Bhattacharyya distance is used. The Bhattacharyya distance would be calculated by [13]
\[ D_B(p,q) = -\ln(BC(p,q)) \]  \hspace{1cm} (4)

\( BC(p,q) \) is Bhattacharyya coefficient, can be calculated by

\[ BC(p,q) = \sum_{x \in X} \sqrt{p(x)q(x)} \]  \hspace{1cm} (5)

\( p(x) \) is the Probability Density Functions (PDF) for \( P \) and \( q(x) \) is the PDF for \( q \). The expression of Bhattacharyya distance \( D_B(X_1, X_2) \) can be derived as follow [13]:

\[ D_B(X_1, X_2) = \frac{1}{4} \left( \frac{\mu_1 - \mu_2}{\sigma_1^2 + \sigma_2^2} \right)^2 + \frac{1}{2} \ln \frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1\sigma_2} \]  \hspace{1cm} (6)

\( D_{BX}(X_1, X_2) \) and \( D_{BY}(Y_1, Y_2) \) present Bhattacharyya distance of the left-right partition and top-bottom partition respectively. The smaller value of Bhattacharyya distance, the higher the similarity.

---

**Figure 5.** Flowchart of proposed method.

---
2.3. Overall Algorithm

For ensure robustness and universality of the algorithm, the proposed algorithm is only utilized when the CU is square. The figure 5 shows the overall algorithm. First, determine whether the CU is square. Then check the texture of the CU. For CU with simple texture, the multi-type partition types will be skipped. For CU with rich texture, the generated coding information is utilized to early skip the horizontal or vertical partition types.

3. Experiments Result

The overall algorithm is implemented in VTM6.2 [14], and different indexes are used to evaluate its performance. Six classes of video sequences as recommended by JVET, the resolution ranging from 416×240 to 3840×2160 are tested, and the value of QP are selected as (22, 27, 32, 37). In these experiments, the coding performance in the All-Intra is measured by three index, Bjøntegaard delta Peak Signal-to-Noise Rate (BD-PSNR) [15], Bjøntegaard bitrate (BD-BR) and the coding time saving. Then the time saving (ΔTS) would be obtained as follows:

\[
\Delta T_S = \frac{T_0 - T_S}{T_0} \times 100\%
\]

where T₀ represent the encoding time of the original reference software and Tₛ represent the encoding time of the proposed method in the encoder.

| Sequence          | BD-PSNR(dB) | BD-BR(%) | ΔTS (%) |
|-------------------|-------------|----------|---------|
| Tango2            | -0.02       | 0.87     | 54.9    |
| FoodMarket4       | -0.02       | 0.58     | 53.5    |
| Campfire          | -0.02       | 0.46     | 51.3    |
| CatRobot1         | -0.04       | 1.57     | 52.8    |
| DaylightRoad2     | -0.03       | 1.71     | 54.4    |
| ParkRunning3      | -0.02       | 1.47     | 49.4    |
| MarketPlace       | -0.01       | 1.16     | 49.7    |
| RitualDance       | -0.02       | 0.93     | 47.5    |
| Cactus            | -0.03       | 1.74     | 50.4    |
| BQTerrace         | 0.00        | 1.92     | 56.5    |
| BasketballDrive   | -0.04       | 1.10     | 52.3    |
| BasketballDrill   | -0.14       | 2.46     | 57.9    |
| BQMall            | -0.03       | 1.40     | 48.2    |
| PartyScene        | -0.02       | 0.87     | 49.9    |
| RaceHorses        | -0.05       | 0.86     | 48.8    |
| BasketballPass    | -0.06       | 1.99     | 56.6    |
| BQSquare          | -0.01       | 1.27     | 50.2    |
| BlowingBubbles    | -0.01       | 1.12     | 50.3    |
| RaceHorses        | -0.06       | 0.61     | 56.0    |
| FourPeople        | -0.07       | 1.38     | 51.5    |
| Johnny            | -0.04       | 1.78     | 54.1    |
| KristenAndSara    | -0.04       | 1.16     | 52.6    |
| **Average**       | **-0.04**   | **1.31** | **52.3**|

The experiments result is shown in Table 1, the encoding time saved by the overall method ranges from 47.5% to 57.9%, with an average of 52.3% while the BDBR increase is 1.31%. Among the test sequences, the biggest loss in coding efficiency is the “BasketballDrill”, with a 2.46% increase in the BD-BR. The proposed method performs consistently well for high resolutions video, such as class A1-A2, and can save 52.7% of the encoding time, while the increase of BD-BR is only 1.11%, which is beneficial to the application and industrialization of ultra-high-definition video. Overall, the overall
algorithm can save coding time while preserving coding efficiency in any test sequence, which demonstrates the overall method has universality and robustness.

4. Conclusion

For VVC intra coding, we proposed an efficient CU partitioning algorithm based on image texture and information generated during the encoding process. The algorithm uses the generated encoding information to determine the texture complexity of the CU, if the CU has simple texture, then the multi-type partition types will be skipped. For further saving coding time, a method of skipping CU partition types based on the distribution of residual coefficients is proposed for CU with rich texture. According to the similarity comparison results of different partition of the current CU, the partial partition types will be skipped. Finally, the proposed method to conduct extensive experiments on VVC reference software-VTM6.2 to test the performance. As the experimental results illustrated, the overall algorithm achieved a saving of about 52.3% coding time with 1.31% BD-BR increase in average, indicating that the algorithm can effectively reduce computational complexity while preserving encoding efficiency.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61571102); Chongqing Basic and Frontier Research Project(cstc2017jcyjXBX0037).

References

[1] G J Sullivan, J Ohm, W Han and T Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
[2] A Segall, V Barontini, J Boyce, J Chen and T Suzuki, “Joint call for proposals on video compression with capability beyond HEVC”,” JVETH1002, Joint Video Exploration Team (JVET). Oct. 2017.
[3] B Bross, J Chen, and S Liu, “Jvet common test conditions and software reference configurations,” Joint Video Experts Team (JVET) ofITU-T SG 16 WP 3 and ISO/IEC JTC l/SC 29/WG 11, vol. JVET-K1001, no. Ljubljana, SI, pp. 10-18, July 2018.
[4] J An, H Huang, and K Zhang. “Quadtree plus binary tree structure integration with JEM tools,” JVET-B0023, Joint Video Exploration Team (JVET). Feb. 2016.
[5] Xin Zhao, Jianle Chen, Marta Karczewicz, Li Zhang, Xiang Li and Wei Jung Chien, “Enhanced multiple transform for video coding,” in Data Compression Conference, 2016, pp. 73-82.
[6] A Said, X Zhao, J Chen, M Karczewicz, W-JChien, and F Zhou, “Position dependent intra prediction combination,” MPEG doc. m37502 and ITU-T SG16 Doc. COM16-C1016, Oct. 2015.
[7] T Fu, H Zhang, F Mu, and H Chen, “Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding,” 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 55-60, Aug. 2019.
[8] M Lei, F Luo, X Zhang, S Wang, and S Ma, “Look-Ahead Prediction Based Coding Unit Size Pruning for VVC Intra Coding,” 2019 IEEE International Conference on Image Processing (ICIP), pp. 4120-4124, sept. 2019.
[9] X Li, F Bossen, J Boyce, K Shring, and V Seregin, “Jvet common test conditions and software reference configurations for sdr video,” document JVET-K1010, 11th JVET meeting: Ljubljana, SI, 2018.
[10] J Hou, D Li, Z Li and X Jiang, "Fast CU size decision based on texture complexity for HEVC intra coding," Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shengyang, 2013, pp. 1096-1099, doi: 10.1109/MEC.2013.6885226.
[11] C Sun, J Su, W Ren, Z Shi and Y Guan, "Motion Vector Analysis Based Early Skip Mode Decision by Bayesian Method for HEVC," 2018 International Conference on Audio, Language
and Image Processing (ICALIP), Shanghai, 2018, pp. 174-178, doi: 10.1109/ICALIP.2018.8455411.

[12] S Cho and M Kim, "Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 9, pp. 1555-1564, Sept. 2013, doi: 10.1109/TCSVT.2013.2249017.

[13] T Kailath, “The divergence and Bhattacharyya distance measures in signal selection,” IEEE Trans Commun Technol 15(1):52–60,1967.

[14] https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftwareVTM/-/tags/VTM-6.2.

[15] G Bjontegaard, “Calculation of average psnr difference between rd-curves,” 13th VCEG-M33 Meeting, Austin, TX, Apr. 2001.