SSIM-Variation-Based Complexity Optimization for Versatile Video Coding

Jielian Lin, Hongbin Lin, Zhichen Zhang, Yiwen Xu, Member, IEEE, and Tiesong Zhao

Abstract—Hitherto, Versatile Video Coding (VVC) has a more magnificent overall performance than High Efficiency Video Coding (HEVC). The Quadtree with Nested Multi-Type Tree (QTMT) coding block structure can substantially enhance video coding quality in VVC. However, the coding gain also leads to a greater coding complexity. Therefore, this letter proposes a Fast Decision Scheme Based on Structural Similarity Index Metric Variation (FDS-SSIMV) to solve this problem. Firstly, the Structural Similarity Index Metric Variation (SSIMV) characteristic among the sub coding units of the splitmode is illustrated. Next, to evaluate the SSIMV value, SSIMV measure strategies are designed for different split modes in this letter. Then, the desired split modes are selected by the SSIMV values. Experimental results show that the proposed method achieves an average encoding Time Saving (TS) and Bjontegaard Delta Bit Rate (BDBR) with 64.74% and 2.79%, respectively, outperforming the benchmarks.

Index Terms—Versatile video coding, inter prediction, supervised contrastive learning, complexity optimization.

I. INTRODUCTION

W
tith the development of video coding technology, the up-to-date video coding standard has been updated to Versatile Video Coding (VVC) [1] by the Joint Video Exploration Team (JVET). The goal of VVC is to reduce the bit rate by 50% than High Efficiency Video Coding (HEVC) [2] at the same quality. To achieve this goal, many new coding techniques have been explored in the newest standard. However, these techniques also lead to a significant increase in the complexity of video coding. Therefore, the complexity optimization of VVC has become one of the most concerned issues of researchers.

For VVC, intra prediction is essential for the block-based coding standard. Many new techniques have been integrated into intra prediction of VVC, including the Quadtree with Nested Multi-Type Tree (QTMT) [3], 67 angular modes, Multiple-Reference Line (MRL) [4], Intra Sub-Partitions (ISP) [5] and Intra Block Copy (IBC) [6], and so forth. These techniques lead to a higher coding complexity, especially for QTMT. The paper [7] has verified that if the unnecessary splits were not tested, the encoding time could save 97.5% on VTM3.0 under the All Intra (AI) configuration. Therefore, it is imperative to decrease the complexity of VVC while keeping the coding efficiency. It is worth noticing that the straightforward scheme of the complexity reduction is to terminate the Coding Unit (CU) partition after testing the Non-Partition (NP) mode or skip some unnecessary split modes (e.g., Quadtree (QT), Horizontal Binary Tree (BTV), Vertical Binary Tree (BT), Horizontal Ternary Tree (TTH), or Vertical Ternary Tree (TT)). As illustrated in Fig. 1.

To date, the predominant complexity optimization methods are to predict the necessary split modes by data-driven approaches [8], [9], [10], [11], [12], [13], [14] and statistical approaches [15], [16], [17], [18], [19], [20]. For data-driven approaches, Yang et al. [9] proposed a learning-based fast CU partition approach for intra-prediction. The textural complexity of the current CU, local texture information, and the context information of neighboring CUs are used as the features to train the decision tree classifiers. Park et al. [10] fed the extracted features (e.g., QT depth, BT/TT depth, block shape ratio) into the designed Lightweight Neural Network (LNN). Based on the trained model, the method can optimize the complexity by terminating the TTH and TTV split modes. Li et al. [11] established a large-scale database and proposed a multi-stage exit CNN model to determine the partition modes. This learning method is adapted to different CU sizes for complexity optimization. Dong et al. [14] proposed an adaptive mode pruning method to eliminate non-promising modes and a mode-dependent termination method to terminate unnecessary intra predictions of remaining depth levels with learning-based classifiers. Reference [13] et al. proposed a multi-information fusion CNN (MF-CNN) model to early terminate the QTMT-based CU partition. Then, by utilizing the CU prediction residuals and the confidence of MF-CNN, a content complexity-based early Merge mode decision
is proposed to skip the inter prediction modes. For statistical approaches, Mai et al. [15] proposed Structural Similarity Index Metric (SSIM) [21]-based fast decision methods, in which Rate Distortions (RD) was utilized to determine the split modes of CUs. Fu et al. [16] optimized the complexity by two steps. Firstly, the RD cost of horizontal binary-tree mode is checked. Then, a Bayesian-based classifier is used to decide whether to skip other split modes. In [17], the Fast Decision Based on Variance (FDV) and Fast Decision Based on Intra Sub-Partition (FD-ISP) strategies are designed to decide whether to skip the binary and ternary partition. Shen et al. [20] developed an early determination of intra mode decision and a bypass strategy for intra prediction on large CU size with texture property and coding information. In summary, most of the above methods only consider the features (e.g., mean, variance, RD cost, and gradient) of the neighbor CUs and the current CU. However, the relationship between the split mode selection and the sub CUs characteristic of the CU has not been investigated.

To solve the above problem, we propose a Fast Decision Scheme Based on Structural Similarity Index Metric Variation (FD-S-SSIMV) in this letter. The contributions of this letter are summarized as follows:

- The asymmetry characteristic is firstly observed and analyzed. Subsequently, a Structural Similarity Index Metric Variation (SSIMV) measure scheme is designed to evaluate the asymmetry characteristic among the sub CUs for the corresponding split modes.
- The FDS-SSIMV method is presented to realize the intra coding complexity reduction. It firstly takes advantage of the asymmetry characteristic of the CU and thus verifies the performance.
- Experimental results evidence that the method with negligible overhead outperforms the benchmarks on different reference softwares in the overall performance.

II. FAST CU PARTITION SCHEME

A. Review and Motivation

1) Review of QTMT Partition: In VVC, the QTMT partition structure has been introduced to enhance the coding performance. The Coding Tree Unit (CTU) firstly performs QT partition as the root node, and then the leaf nodes are set to test all permitted partition modes (e.g., NP, QT, BTH, BTV, TTH, or TTV, as illustrated in Fig. 1). After recursive coding, the smallest RD cost of the CU can be obtained. It can be expressed as:

\[
J_{CU} = \min_{m \in M} \{ D_m + \lambda \times R_m \},
\]

where \( J_{CU} \) is the minimum RD cost of the current CU. \( m \) and \( M \) are the index of the current test partition mode and the number of all permitted partition modes for the current CU. \( D_m \) and \( R_m \) are distortion and bit rate of the \( m \)-th partition mode of the CU, respectively. \( \lambda \) is the Lagrangian parameter. As illustrated above, the partition mode with the smallest RD cost is the best partition scheme. Therefore, the complexity optimization method can accelerate video coding by skipping the split modes with a higher RD cost to accelerate the coding effectively.

2) Motivation: According to (1), the complexity optimization method needs to decide the most necessary split mode to be tested by terminating the partition or skipping some split modes. However, as shown in Fig. 1, the best partition mode of parent CU has many different sub best partition modes. It implies that the CU has the asymmetry characteristic under the split modes.

To evaluate the asymmetry of the CU, the SSIM metric [21] is adapted. This experiment is carried out on the 32×32 CU. The typical 32×32 CU is divided into sixty-four 4×4 non-overlapped blocks. The SSIM values of these non-overlapped blocks are calculated with the corresponding original CU blocks and predicted CU blocks. The detailed formula is:

\[
SSIM(o, p) = \frac{(2\mu_o\mu_p + c_1)(2\sigma_o\sigma_p + c_2)}{\left(\mu_o^2 + \mu_p^2 + c_1\right)\left(\sigma_o^2 + \sigma_p^2 + c_2\right) + c_1},
\]

where \( o \) and \( p \) are the original CU and predicted CU with NP mode, respectively. \( \mu_o \) and \( \mu_p \) are the average values of the original CU and predicted CU. \( \sigma_o \) and \( \sigma_p \) are the variance values of the original CU and predicted CU. \( c_1 \) and \( c_2 \) are the constant values. They are set as \( c_1 = (k_1L)^2 \) and \( c_2 = (k_2L)^2 \). \( k_1 \) is set as 0.01. \( k_2 \) is set as 0.03. \( L \) is set as \( 2^{10} - 1 \). The size of the Gaussian filter is set as 4×4.

After that, according to the split modes, SSIM values of non-overlapping blocks are classified into different categories (QT1, QT2, QT3, and QT4 for QT split mode, BTH1 and BTH2 for BTH split mode, BTV1 and BTV2 for BTV split mode, TTH1, TTH2, and TTH3 for TTH split mode and TTV1, TTV2, and TTV3 for TTV split mode). As illustrated in Fig. 2, it indicates the SSIMV characteristic of the CU. Especially for TTV split mode, the average SSIM values of the sub CUs show a large variation. This characteristic indicates that the TTH1 sub CU of this split mode has a significant difference between the original CU and predicted CU, and other sub CUs have a higher similarity than the other split mode. Therefore, it indicates this split mode can split the CU more precisely and have a more chance to obtain the smallest RD cost. Then, the degree of the asymmetry of the CU under different split modes may indicate the necessary split modes of the CU.

B. SSIM-Variation Calculation Scheme

Based on the above analysis, we designed the calculation schemes to evaluate the asymmetry of the CU for different split modes. The notations of the SSIMV value for the split modes are...
expressed as $V_{QT}$, $V_{BTH}$, $V_{BTV}$, $V_{TTH}$ and $V_{TTV}$, respectively. The SSIMV values of these partition modes are calculated as follows.

Firstly, $V_{QT}$ can be obtained by:

$$V_{QT} = \frac{(D_{A1} + D_{A2} + D_{A3} + D_{A4})}{4},$$  \hspace{1cm} (3)

where $D_{A1}$, $D_{A2}$, $D_{A3}$ and $D_{A4}$ are the difference in the SSIM values of the neighbor sub CUs. They are expressed as:

$$\begin{align*}
D_{A1} &= |S_{a1} - S_{a2}|, \\
D_{A2} &= |S_{a1} - S_{a3}|, \\
D_{A3} &= |S_{a2} - S_{a4}|, \\
D_{A4} &= |S_{a3} - S_{a4}|,
\end{align*}$$  \hspace{1cm} (4)

where $S_{a1}$, $S_{a2}$, $S_{a3}$ and $S_{a4}$ are the SSIM values of the corresponding sub CUs in Fig. 3(a), respectively. The size of the Gaussian filter used for SSIM calculation is set as $11 \times 11$ for the size of sub-CUs greater than 11. Otherwise, it is set as $4 \times 4$.

Similarly, $V_{BTH}$ and $V_{BTV}$ can also be obtained by:

$$V_{BTH} = V_{BTV} = |S_{b1} - S_{b2}|,$$  \hspace{1cm} (5)

where $S_{b1}$ and $S_{b2}$ are the SSIM values of the corresponding sub CUs in Fig. 3(b) and (c), respectively.

Besides, $V_{TTH}$ and $V_{TTV}$ can be calculated by:

$$V_{TTH} = V_{TTV} = \frac{(D_{C1} + D_{C2} + D_{C3})}{4},$$  \hspace{1cm} (6)

where $D_{C1}$, $D_{C2}$ and $D_{C3}$ are the difference in the SSIM values of neighbor sub CUs. The sub CUs of the TTH and TTV split modes, which are used to evaluate the SSIMV value, are designed as Fig. 3(d) and (e). $D_{C1}$, $D_{C2}$ and $D_{C3}$ are expressed as:

$$\begin{align*}
D_{C1} &= |S_{c1} - S_{c2}|, \\
D_{C2} &= |S_{c1} + S_{c4} - S_{c2} - S_{c3}|, \\
D_{C3} &= |S_{c3} - S_{c4}|,
\end{align*}$$  \hspace{1cm} (7)

where $S_{c1}$, $S_{c2}$, $S_{c3}$ and $S_{c4}$ are the SSIM values of the corresponding sub CUs in Fig. 3(d) and (e), respectively.

Based on the SSIMV feature, we propose a scheme called FDS-SSIMV. The scheme can select the necessary test split modes in VVC. Additionally, the SSIMV characteristic of the split modes is used to evaluate the asymmetry of the CU. The higher SSIMV value of the split mode indicates that the sub CUs of the split mode have more significant variations. To a certain extent, the phenomenon implies that the necessary split mode of the CU has a higher SSIMV value. Therefore, we sort the SSIMV values of all the split modes to determine the most needed testing split mode. The number of skipping split modes $N$ is defined as $\lceil \frac{N_p}{4} \rceil$. $N_p$ is the permitted number of split modes of the CU. It is noteworthy that skipping the QT partition may lead to a huge loss in encoding performance. Therefore, our fast algorithm has been designed not to skip the QT split mode. This design adjusts the number of skipped split modes to balance the coding quality and efficiency.

Combined with the above analysis, the flowchart of the proposed method is shown in Fig. 4. In this method, the optimization is designed for the CU size larger than or equal to $8 \times 8$ CU and the luma channel of the CU. Then, the SSIMV values of the permitted split modes are calculated. After that, the testing stack (ComprCUCtxList) is updated with FDS-SSIMV. Based on this updated ComprCUCtxList, the requisite split modes are tested as the original reference software.

### III. EXPERIMENTS

The proposed method is integrated into VTM7.0 and VTM17.0 to verify its effectiveness. This experiment is tested on the mandatory video sequences of the Common Testing Coding (CTC) JVET-N1010 [22] under the AI configuration and Quantization Parameters (QPs) 22, 27, 32, and 37. Furthermore, the performances of the deep-learning-based [11] and statistical complexity optimization methods [9], [16] are also compared in this section. There are denoted as Yang–TCSVT’20 [9], Li–TIP’21 [11] and Fu–ICME’19 [16], respectively.

The Bjontegaard Average Bit Rate (BDBR) [23], TS, and Overall Performance (OP) metrics are adopted to verify the performance of the proposed method. To evaluate the overall performance of the proposed method, OP is calculated by $TS_{\text{average}} / BDBR_{\text{average}}$. $TS_{\text{average}}$ and $BDBR_{\text{average}}$ are the average value of TS and BDBR, respectively. The higher OP indicates a better performance of the method.

In Table I, the performances of the proposed method and benchmarks are presented. For VTM7.0, the average BDBR
of the proposed method is 2.79%, which outperforms the Fu-ICME’19 and Li-TIP’21 methods by 0.65% and 0.5%, respectively. For TS performance, the proposed method reduces 54.82% ∼ 70.61% TS. The average TS of the proposed method is 64.74%, which is superior to the average TS of the Fu-ICME’19, Yang-TCSVT’20, and Li-TIP’21 methods with 57.8%, 54.32%, and 64.17%, respectively. Although the BDBR of the Yang-TCSVT’20 method is competitive with the proposed method, the average TS is lower than the proposed method by 10.42%. For the OP metric, the proposed method achieves 23.20 values, which outperforms the OP values of Fu-ICME’19, Yang-TCSVT’20, and Li-TIP’21 methods with 16.80, 20.42, and 19.50, respectively. For VTM17.0, the proposed method achieves the average TS and BDBR with 63.09% and 2.81%, respectively. In addition, the OP metric with 22.45 also outperforms the benchmarks. In other words, the method has a guaranteed generalization performance.

To further evaluate the method’s performance, we also calculate the overhead of the proposed method in the original reference software and the integrated software with our method. The overhead

\[ \text{OH}_{\text{pro}} = \frac{1}{4} \sum_{Qp \in \{22, 27, 32\}} T_{\text{ssim}}(Qp_i)/T_{\text{pro}}(Qp_i) \times 100\%, \]

where \( T_{\text{ssim}}(Qp_i) \) is the time cost of the proposed method in running the integrated software to code the sequences. In Table I, the average overhead of the \( \text{OH}_{\text{pro}} \) is 0.6% and 0.25% for the reference software VTM7.0 and VTM17.0, respectively. The overhead of the proposed method is decreased than the 3.67% overhead of the Li-TIP’21 method [11]. In other words, the method has a guaranteed generalization performance.

TABLE I

| Sequences | Fu-ICME’19 (VTM7) | Yang-TCSVT’20 (VTM7) | Li-TIP’21 (VTM7) | Proposed (VTM7) | Proposed (VTM7) |
|-----------|-------------------|----------------------|-----------------|-----------------|-----------------|
| Class     | TS (%)            | BDBR (%)             | TS (%)          | BDBR (%)        | BDBR (%)        |
| A1        | FoodMarked        | 53.32                | 3.49             | 53.71           | 2.47            |
|           |                   |                      |                 |                 |                 |
| A2        | DaylightRoad                  | 60.83                | 2.80             | 62.81           | 2.48            |
|           |                   |                      |                 |                 |                 |
| B         | BasketballDrive            | 57.95                | 3.75             | 58.97           | 3.13            |
|           |                   |                      |                 |                 |                 |
| C         | BasketballDrive            | 52.91                | 4.19             | 53.34           | 3.54            |
|           |                   |                      |                 |                 |                 |
| D         | BasketballDrive                  | 55.27                | 3.88             | 55.34           | 3.54            |
|           |                   |                      |                 |                 |                 |
| E         | BasketballDrive                  | 56.62                | 4.19             | 56.91           | 3.78            |
|           |                   |                      |                 |                 |                 |
|          |                    |                      |                 |                 |                 |
| OP        | 16.80              | 20.42                | 19.50           | 22.45           | 22.45           |

In this letter, we firstly observe and analyze the asymmetry characteristic of the CUs with SSIM. Then, the SSIM schemes are designed to evaluate the asymmetry characteristic for the corresponding split modes. By utilizing the SSIM characteristic, the FDS-SSIMV scheme is proposed to optimize the complexity of VVC video coding in this letter. Experimental results indicate that the proposed method achieves a light overhead, high TS, and competitive BDBR performance method. Moreover, compared with the benchmarks, the overall performance of the proposed method is superior to them on different reference softwares.

IV. CONCLUSION

In this letter, we firstly observe and analyze the asymmetry characteristic of the CUs with SSIM. Then, the SSIM schemes are designed to evaluate the asymmetry characteristic for the corresponding split modes. By utilizing the SSIM characteristic, the FDS-SSIMV scheme is proposed to optimize the complexity of VVC video coding in this letter. Experimental results indicate that the proposed method achieves a light overhead, high TS, and competitive BDBR performance method. Moreover, compared with the benchmarks, the overall performance of the proposed method is superior to them on different reference softwares.

REFERENCES

[1] F. Bossen, X. Li, and K. Suehring, “JVET AHG report: Test model software development (AHG3),” Joint Video Experts Team of ITU-T SG 16 WP 16 SG 3 and ISO/IEC JTC 1/SC 29/WG 11, document JVET-Q0003-V1, Geneva, Switzerland, Jan. 2020.

[2] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] B. Bossen, “Versatile video coding (Draft 1),” Joint Video Experts Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JVET-J1001, Geneva, Switzerland, Apr. 2018.

[4] L. Zhao, X. Zhao, X. Li, and S. Liu, “Further investigations on multi-line intra prediction,” Joint Video Experts Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JVET-J0065, Geneva, Switzerland, Apr. 2018.

[5] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne, and D. Menard, “Complexity reduction opportunities in the future VVC intra encoder,” in Proc. IEEE Int. Workshop Multimedia Signal Process., Kuala Lumpur, Malaysia, 2019, pp. 1–6.
[8] L. Zhu, Y. Zhang, S. Kwong, X. Wang, and T. Zhao, “Fuzzy SVM-based coding unit decision in HEVC,” *IEEE Trans. Broadcast.*, vol. 64, no. 3, pp. 681–694, Sep. 2018.

[9] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, “Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1668–1682, Jun. 2020.

[10] S. Park and J. Kang, “Fast multi-type tree partitioning for versatile video coding using a lightweight neural network,” *IEEE Trans. Multimedia*, vol. 23, pp. 4388–4399, 2021.

[11] T. Li, M. Xu, R. Tang, Y. Chen, and Q. Xing, “DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC,” *IEEE Trans. Image Process.*, vol. 30, pp. 5377–5390, 2021.

[12] G. Fu, L. Shen, H. Yang, X. Hu, and P. An, “Fast intra coding of high dynamic range videos in SHVC,” *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1665–1669, Nov. 2018.

[13] Z. Pan, P. Zhang, B. Peng, N. Ling, and J. Lei, “A CNN-based fast inter coding method for VVC,” *IEEE Signal Process. Lett.*, vol. 28, pp. 1260–1264, 2021.

[14] X. Dong, L. Shen, M. Yu, and H. Yang, “Fast intra mode decision algorithm for versatile video coding,” *IEEE Trans. Multimedia*, vol. 24, pp. 400–414, 2022.

[15] Z. Mai, C. Yang, and S. Xie, “Improved best prediction mode(s) selection methods based on structural similarity in H.264 I-frame encoder,” in *Proc. IEEE Int. Conf. Syst. Man, Cybern.*, Waikoloa, HI, USA, vol. 3, 2005, pp. 2673–2678.

[16] T. Fu, H. Zhang, F. Mu, and H. Chen, “Fast CU partitioning algorithm for H.266/VVC intra-frame coding,” in *Proc. IEEE Int. Conf. Multimedia Expo*, Shanghai, China, 2019, pp. 55–60.

[17] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, “Fast partitioning decision scheme for versatile video-coding intra-frame prediction,” in *Proc. IEEE Int. Symp. Circuits Syst.*, Seville, Spain, 2020, pp. 1–5.

[18] M. Tang, X. Chen, J. Gu, Y. Han, J. Wen, and S. Yang, “Accelerating HEVC encoding using early-split,” *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 209–213, Feb. 2018.

[19] J. Gu, M. Tang, J. Wen, and Y. Han, “Adaptive intra candidate selection with early depth decision for fast intra prediction in HEVC,” *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 159–163, Feb. 2018.

[20] L. Shen, Z. Zhang, and Z. Liu, “Effective CU size decision for HEVC intracoding,” *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232–4241, Oct. 2014.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[22] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, “VTM common test conditions and software reference configurations for SDR video,” Joint Video Experts Team of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JVET-N1010, Geneva, Switzerland, Mar. 2019.

[23] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” ITU - Telecommunications Standardization Sector document VCEG-M33, Austin, TX, USA, 2001.

[24] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *Proc. Conf. Rec. Asilomar Conf. Signals Syst. Comput.*, vol. 2, 2003, pp. 1398–1402.