Block-level Lagrange multiplier adaptation based on distortion propagation factors

Hongwei Guo[1, 2], Yipeng Liu[2], Lei Luo[2,3], Chengbiao Fu[4], and Ce Zhu[2]
[1]School of Engineering, Honghe University, Mengzi, Yunnan, China
[2]School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China
[3]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
[4]Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China
ghw202@foxmail.com, yipengliu@uestc.edu.cn, luolei@cqupt.edu.cn, fcb@kust.edu.cn, eczhu@uestc.edu.cn

Abstract—**Temporally dependent rate-distortion optimization methods typically adjust the Lagrange multiplier at the coding tree unit (CTU) level to optimize bit resource allocation, thereby enhancing the video encoder's rate-distortion (R-D) performance. However, such methods can be too coarse-grained, especially in Versatile Video Coding (VVC), where the CTU size is increased to 128×128. To more precisely address the limitations of CTU-level Lagrange multiplier adaptation, this paper introduces a block-level Lagrange multiplier adaptation method. This method enables more refined bit resource allocation by adjusting optimization targets for different regions within a CTU, thereby further improving the compression performance of the video encoder. Experimental results using the VVC reference software VTM-23.4 show that the proposed method generally outperforms state-of-the-art algorithms. Specifically, compared to the original VTM-23.4, the proposed algorithm achieves average R-D performance gains of 4.28% and 4.47% under low-delay B-slice and P-slice configurations, respectively.**

I. INTRODUCTION

Driven by the proliferation of video capture devices, the volume of video data has been growing explosively. Since the 1990s, international standards-setting organizations have developed and released a series of video coding standards to reduce the cost of video data storage and transmission[1]. Versatile Video Coding (VVC), finalized in July 2020, achieves roughly a 50% bit rate reduction at the same subjective video quality compared to its predecessor High Efficiency Video Coding (HEVC) [2]. Like its predecessors, VVC employs a block-based hybrid coding framework that includes modules for prediction, transform, quantization, sample adaptive offset, in-loop filtering, and entropy coding. In video encoders such as HEVC and VVC, the coding tree unit (CTU) is the basic unit for independent mode decisions; notably, VVC increased the maximum CTU size from 64×64 to 128×128 pixels [3].

Rate-distortion optimization (RDO) [4] is a key technique in video encoding, balancing the coding bit rate and reconstruction distortion to determine the optimal coding mode for each unit. A flexible RDO algorithm can significantly enhance compression efficiency. Accordingly, each new video coding standard has spurred research into improved RDO methods. For instance, in HEVC's hierarchical coding structure, several adaptive quantization parameter (QP) cascading schemes have been proposed. Under low-delay configurations, these methods dynamically adjust frame-level QPs based on content variations [5-7], and analogous QPC techniques have been developed for random access configurations [8, 9]. By analyzing inter-frame rate-distortion (R-D) dependencies, such methods assign appropriate QPs to each frame in a group of pictures (GOP) and derive the corresponding Lagrange multiplier (λ) from the QP. Each frame is then encoded using a uniform λ–QP pair across all its CTUs. Beyond frame-level QP adaptation, an approach in [10] employs a pre-encoding analysis that uses original frames as references to estimate temporal distortion propagation, thereby enabling adaptive QP selection at the CTU level. This CTU-level QP cascading scheme achieved better R-D performance than prior frame-level QP cascading methods.

In earlier research, Yang et al. [11] investigated temporally dependent RDO in H.264/ Advanced Video Coding (AVC) by developing a source-distortion temporal propagation model. This model adaptively adjusted the Lagrange multiplier for each macroblock, yielding significant coding efficiency gains. Subsequent studies extended similar temporal propagation models to HEVC's low-delay [12] and random access [13] configurations, likewise achieving R-D performance improvements. Building on these advances, several RDO enhancements have been explored for VVC. Xu et al. [14] introduced a CTU-level λ adaptation algorithm for VVC encoders, while Chen et al. [15] presented a two-pass encoding scheme for adaptive QP selection in VVC random access coding. Enhorn et al, [16] introduced a block importance mapping (BIM), calculating from the temporal filter [17], and then proposed an adaptive QP offset at CTU level. The BIM algorithm has been adopted by the Joint Video Experts Team (JVET) [18] and integrated into the VVC test model (VTM) as

979-8-3315-7206-8/25/$31.00 ©2025 IEEE

an optional encoding optimization tool for performance evaluation. Another line of work focuses on explicitly quantifying temporal R-D dependency through distortion propagation factors (DPF). In [19], DPF were introduced to adaptively adjust coding parameters, yielding substantial R-D performance gains for HEVC. This concept was further refined in [20], which developed an improved DPF calculation for VVC low-delay hierarchical coding. The DPF-based λ adaptation at CTU-level achieved significant R-D performance improvements in VVC and has been adopted by the JVET [21] for integration into the VVC reference software VTM-23.4 and subsequent versions. Recently, studies have continued to advance RDO techniques for VVC, including intra-frame bit allocation based on significance detection [22] and rate control methods utilizing Lagrange multiplier adjustment for small bit fluctuations [23].

In summary, the above RDO optimization techniques primarily improve coding efficiency by adjusting the Lagrange multiplier and QP at either the frame level or the CTU level. However, CTU-level λ adaptation applies a single λ value uniformly to an entire CTU based on its average R-D significance. This approach can be too coarse-grained, especially in VVC, where the CTU size is larger (128×128). Within such a large CTU, different regions may exhibit varying importance in terms of R-D trade-offs. To more precisely address the shortcomings of CTU-level adaptation, this paper introduces a block-level Lagrange multiplier adjustment scheme. By fine-tuning λ at a smaller block granularity within each CTU, the proposed method enables more precise temporally dependent R-D optimization and further improves the compression performance of the VVC encoder beyond what CTU-level adaptation can achieve.

The remainder of this paper is organized as follows. Section II reviews temporally dependent RDO at CTU-level. Section III details the proposed block-level Lagrange multiplier adaptation method. Section IV presents the experimental results, and Section V concludes the paper.

II. TEMPORALLY DEPENDENT RDO AT CTU-LEVEL

Traditional RDO methods focus on optimizing the R-D performance for each CTU independently. In contrast, temporally dependent RDO incorporates the temporal R-D dependencies between frames to further enhance coding efficiency. Within the current block-based hybrid coding framework, temporally dependent RDO can be implemented using the following equation:

$$\lambda_n = \frac{\lambda_g}{1+\partial\left(\sum_{i=n+1}^{N} D_i\right)\Big/\partial D_n}, \qquad n = 1,2,\ldots,N \tag{1}$$

where λ_g represents the global Lagrange multiplier, λ_n denotes the Lagrange multiplier for the n-th pixel block, and D_n is its corresponding coding distortion. Additionally, N is the total number of pixel blocks in the encoded video. The distortion term in the equation defines the rate at which the distortion of subsequent video segments changes due to fluctuations in the current block's distortion, referred to as the DPF, as follows:

$$\phi_n = \partial\left(\sum_{i=n+1}^{N} D_i\right)\Big/\partial D_n \tag{2}$$

Therefore, the key to temporally dependent RDO lies in how to obtain the DPF.

A. Estimation of DPF

Due to the need to use distortion variation for calculating the DPF defined in (2), it is not possible to directly obtain the DPF during the video encoding process based solely on this equation. Therefore, in our previous work [20], we derived a formula for the DPF that involves only motion-compensated prediction (MCP) errors and coding distortion, based on (2) and R-D theory. Furthermore, we proposed a pre-encoding-based DPF estimation method for VVC low-delay hierarchical coding in [20]. Specifically, to obtain the coding distortion and MCP error for each pixel block in the frame, a pre-encoding step is first performed on the frame to be encoded. The QP value for the pre-encoding is set to the maximum QP used in a GOP. For example, when the input QP of the encoder is 32, the maximum QP value in the GOP would be 41. To minimize the computational burden, only fixed 32×32 pixel blocks with integer-pixel inter-frame prediction are used, and no post-processing tools, such as loop filters and sample adaptive offsets, are applied. Compared to the full encoding process, which involves time-consuming motion estimation and mode selection, the computational cost of pre-encoding is relatively minimal. After pre-encoding, the DPF for each 32×32 pixel block in the frame is computed as follows:

$$\phi_n = \sum_{k=1}^{L_n}\left(\frac{D_n}{D_n^{\mathrm{MCP}}}\right)^k \tag{3}$$

where D_n and D_n^{MCP} represent the coding distortion of the n-th pixel block and its MCP error, both measured by mean squared error (MSE). L_n is set to 8 for key frames and 1 for non-key frames.

B. Lagrange Multiplier Adaptation at CTU-Level

As mentioned earlier, in both HEVC and VVC encoders, the CTU serves as the basic unit for independent mode decisions. Therefore, temporally dependent RDO methods typically enhance the R-D performance by adjusting parameters at the CTU-level, such as the Lagrange multiplier or QP. In [20], a weight coefficient of 32×32 pixel blocks is introduced based on the DPF estimated by (3), as follows:

$$\omega_n^o = \frac{1}{1+\phi_n} \tag{4}$$

Since the weight coefficient ω_n^{o} is always less than 1, each weight is normalized by dividing it by the mean of all weights, resulting in new weights centered around 1, as shown in the following equations:

$$\omega_{\mathrm{ave}} = \frac{1}{N_{\mathrm{blk}}} \sum_{n=1}^{N_{\mathrm{blk}}} \omega_n^o \tag{5}$$

$$\omega_n = \frac{\omega_n^o}{\omega_{\mathrm{ave}}} \tag{6}$$

where N_{blk} represents the number of 32×32 pixel blocks in the frame, and ω_{ave} is the arithmetic mean of the weight coefficients in the frame to be encoded. Additionally, ω_n is the normalized weight coefficient.

Next, based on the pixel blocks covered by each CTU, the Lagrange multiplier at CTU-level is calculated using the following equation:

$$\lambda_{\mathrm{CTU}} = \left(\frac{1}{M_{\mathrm{CTU}}} \sum_{n=1}^{M_{\mathrm{CTU}}} \omega_n \right) \cdot \lambda_{\mathrm{F}} \tag{7}$$

where M_{CTU} represents the number of blocks included in the current CTU of the frame being encoded, and λ_{F} is the frame-level Lagrange multiplier for the current frame.

III. PROPOSED METHOD

The CTU-level Lagrange multiplier adaptation method adjusts the Lagrange multiplier based on the average weight coefficient of the pixel blocks within the current CTU, allowing the encoder to independently determine the optimal coding mode. This approach is convenient for implementation in current block-based hybrid video encoders. However, it does not fully exploit the R-D dependency differences across various regions within the CTU. This limitation becomes particularly apparent in VVC encoders, where the use of larger CTU sizes makes the CTU-level Lagrange multiplier adaptation relatively coarse. To address this, this section proposes a block-level Lagrange multiplier adaptation method that offers more refined bit resource allocation by adjusting optimization targets for different regions within the CTU, thereby further enhancing the compression performance of the video encoder. This method refines the CTU coding mode decision process, and the specific steps are outlined as follows:

First, for a coding unit (CU) at depth 0, the CTU-level Lagrange multiplier λ_{CTU}, as calculated by (7), is used for deciding the mode of the current CU. This includes evaluating prediction modes such as Skip, Merge, Advanced Motion Vector Prediction (AMVP), and intra prediction. The R-D cost for the optimal prediction mode at depth 0, denoted as $J_{depth=0}$, is given by:

$$J_{depth=0} = D + \lambda_{\mathrm{CTU}} \cdot R \tag{8}$$

where D and R represent the reconstruction distortion and bit rate associated with the best prediction mode of the CU at depth 0.

Next, for CUs at depth 1, the block-level Lagrange multiplier λ_{Block} is used for both prediction mode selection and further partitioning decisions. This allows for the determination of the optimal coding mode for each CU at depth 1, including both partitioning and prediction modes. The block-level Lagrange multiplier λ_{Block} is computed as follows:

$$\lambda_{\mathrm{Block}} = \left(\sum_{n=1}^{M_{\mathrm{CU}}} \omega_n \cdot \beta_n \Big/ \sum_{n=1}^{M_{\mathrm{CU}}} \beta_n \right) \cdot \lambda_{\mathrm{F}} \tag{9}$$

where n indicates the index of the pixel blocks covered by a CU at depth 1. The normalized weight coefficient for the n-th block, ω_n, is calculated according to (6). β_n denotes the proportion of the n-th pixel block that is contained within the CU. For example, if half of the pixels in the n-th block are included, then $\beta_n = 0.5$. M_{CU} is the total number of pixel blocks covered by the CU.

After the optimal coding modes for all CUs at depth 1 have been determined, the overall R-D cost at depth 1, denoted as $J_{depth=1}$, is calculated as:

$$J_{depth=1} = \widehat{D} + \lambda_{\mathrm{CTU}} \cdot \widehat{R} \tag{10}$$

where $\widehat{D}$ and $\widehat{R}$ are the total reconstruction distortion and bit rate, respectively, when the optimal coding modes for all CUs at depth 1 are selected.

Finally, the coding mode corresponding to the smaller value between $J_{depth=0}$ and $J_{depth=1}$ is selected as the final coding mode for the current CTU, thus completing its encoding process. It is important to note that while the same Lagrange multiplier is used when calculating $J_{depth=0}$ and $J_{depth=1}$, different Lagrange multipliers are applied during the mode decision process for the CU at depth 0 and the CUs at depth 1.

IV. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed method, experiments were conducted using the VVC reference software VTM-23.4. As background, the BIM [16] algorithm was adopted by JVET and integrated into VTM-16.0 and subsequent versions, as outlined in JVET-Y0077 [18]. Similarly, the DPF [20] algorithm was adopted and incorporated into VTM-23.4, as specified in JVET-AH0078 [21]. Consequently, four encoders were tested: the original VTM-23.4, the BIM [16] algorithm, the DPF [20] algorithm, and the proposed method.

The experimental setup followed the VTM Common Test Conditions (CTC) for SDR video [24] as specified by JVET. Specifically, the Low-delay B (LB) and Low-delay P (LP)

TABLE I
BDBR for Y Component of Different Methods Compared to Original VTM-23.4 under LB and LP Configurations

Class	Sequence	LB Configuration			LP Configuration		
		BIM [16]	DPF [20]	Proposed	BIM [16]	DPF [20]	Proposed
Class B	BasketballDrive	-3.10%	-3.57%	-3.50%	-2.78%	-3.52%	-3.40%
	BQTerrace	-1.41%	-0.24%	-0.35%	-0.94%	0.61%	0.84%
	Cactus	-2.72%	-3.61%	-3.84%	-2.19%	-3.10%	-3.19%
	MarketPlace	-5.21%	-4.74%	-4.59%	-5.15%	-4.84%	-4.69%
	RitualDance	-3.47%	-6.51%	-6.71%	-3.38%	-6.81%	-7.07%
Class C	BasketballDrill	-5.80%	-12.61%	-14.44%	-5.80%	-13.18%	-15.28%
	BQMall	-2.37%	-2.28%	-2.17%	-2.16%	-2.27%	-2.29%
	PartyScene	-3.42%	-7.33%	-8.72%	-2.98%	-7.20%	-8.60%
	RaceHorses	-1.26%	1.38%	2.24%	-0.71%	1.40%	2.38%
Class D	BasketballPass	-2.17%	-2.74%	-4.30%	-1.81%	-2.90%	-4.37%
	BlowingBubbles	-2.53%	-2.33%	-2.95%	-2.13%	-2.41%	-3.06%
	BQSquare	-0.65%	0.29%	0.70%	-1.19%	0.01%	0.28%
	RaceHorses	-0.95%	-0.83%	-0.75%	-0.57%	-0.88%	-0.86%
Class E	FourPeople	-0.75%	-4.82%	-5.95%	-1.48%	-4.92%	-5.94%
	Johnny	0.14%	-1.68%	-1.55%	0.15%	-1.19%	-1.65%
	KristenAndSara	0.64%	-3.16%	-3.67%	0.26%	-3.41%	-3.74%
Class F	ArenaOfValor	-5.64%	-9.07%	-9.94%	-5.96%	-9.90%	-10.94%
	BasketballDrillText	-4.86%	-11.06%	-12.90%	-5.13%	-11.80%	-13.60%
	SlideEditing	2.30%	1.98%	2.47%	1.94%	2.28%	2.49%
	SlideShow	-2.43%	-5.18%	-4.68%	-2.11%	-6.60%	-6.70%
Average	Class B	-3.18%	-3.74%	-3.80%	-2.89%	-3.53%	-3.50%
	Class C	-3.21%	-5.21%	-5.77%	-2.91%	-5.31%	-5.95%
	Class D	-1.58%	-1.40%	-1.82%	-1.43%	-1.55%	-2.00%
	Class E	0.01%	-3.22%	-3.73%	-0.36%	-3.17%	-3.78%
	Class F	-2.66%	-5.83%	-6.26%	-2.81%	-6.50%	-7.19%
	Overall	**-2.28%**	**-3.91%**	**-4.28%**	**-2.21%**	**-4.03%**	**-4.47%**
Encoding Time		100%	104%	104%	100%	104%	104%
Decoding Time		100%	100%	100%	100%	100%	100%

configurations were evaluated, with all encoder parameters set according to the default settings in "encoder_lowdelay_vtm.cfg" and "encoder_lowdelay_P_vtm.cfg." In line with the CTC, the test set for LB and LP configurations included all 20 video sequences with varying content characteristics and resolutions from Classes B, C, D, E, and F. It is noteworthy that Class F consists of screen content videos, which were encoded with the additional "classF.cfg" profile as required by the CTC. For each sequence, four QP values (22, 27, 32, and 37) were used. These values serve as the base QPs, from which the actual QP for each frame is derived in VTM-23.4. When the BIM [16] algorithm is enabled, the QP values for each CTU are derived according to the BIM mechanism, replacing the standard QPs in VTM-23.4. Similarly, when the DPF [20] algorithm is enabled, QP values for each CTU are indirectly determined based on DPF and substitute the default QPs.

To demonstrate the R-D performance of the proposed algorithm, the Bjøntegaard Delta Bit Rate (BDBR) metric was employed, with the original VTM-23.4 serving as the reference. The BDBR measures the bit rate savings of the test method compared to the benchmark at equivalent coding quality, where quality is evaluated by the PSNR of the Y component in the decoded videos. A positive BDBR indicates a performance loss, whereas a negative value reflects a performance improvement. In addition, the computational complexity of each algorithm was assessed in terms of encoding and decoding time, expressed as a percentage relative to the benchmark encoder.

Table I summarizes the experimental results for each test sequence under both LB and LP configurations. With the LB configuration, BIM [16] and DPF [20] achieve average R-D performance gains of 2.28% and 3.91%, respectively. The proposed method further improves the average R-D performance to 4.28%, representing a 0.37% gain over DPF [20]. Under the LP configuration, BIM [16] and DPF [20] yield average gains of 2.21% and 4.03%, while the proposed method achieves 4.47%, which is 0.44% higher than DPF [20]. For Classes C, D, E, and F, the proposed method consistently outperforms DPF [20] in both configurations. In contrast, for Class B, the improvement over DPF [20] is not significant. This is mainly because Class B comprises 1920×1080 resolution sequences, and for high-resolution videos, the block-level weight coefficients used in the proposed method are almost identical to the CTU-level coefficients in DPF [20], leading to minimal differences in R-D performance. Additionally, for certain sequences , such as BasketballDrive and MarketPlace, the proposed method performs worse than DPF [20] due to the limited accuracy of the estimated DPF for 32×32 pixel blocks. In such cases, the CTU-level weighting, by averaging over more blocks, can partially alleviate the impact of DPF estimation inaccuracies. Regarding computational complexity, BIM [16] introduces no additional overhead, whereas both DPF [20] and the proposed method rely on pre-encoding for DPF estimation and therefore increase the encoding time by approximately 4%.

V. CONCLUSIONS

In this paper, we proposed a novel block-level Lagrange multiplier adaptation method based on DPFs for VVC. Unlike conventional CTU-level adaptation approaches, the proposed method leverages the R-D dependency differences across various regions within a CTU to achieve more precise allocation of coding resources. By introducing block-level weight coefficients, the algorithm enables finer control over the RDO process, resulting in improved compression performance, particularly for sequences with lower resolutions or more complex content. Experimental results demonstrate that the proposed method generally outperforms state-of-the-art algorithms, while maintaining reasonable computational complexity.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 62461023 and 62061015; in part by the Key Industry Science and Technology Service Project of Yunnan Provincial Universities under Grant FWCY-ZNT2024025; and in part by the Fund for Reserve Talents of Young and Middle-aged Academic and Technical Leaders of Yunnan under Grant 202205AC160055.

REFERENCES

[1] H. Guo, Y. Zhou, H. Guo, Z. Jiang, T. He, and Y. Wu, "A Survey on Recent Advances in Video Coding Technologies and Future Research Directions," *IEEE Trans. Broadcast.,* vol. 71, no. 2, pp. 666 - 671, Apr., 2025.

[2] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)," *Proc. IEEE,* vol. 109, no. 9, pp. 1463 - 1493, Sep., 2021.

[3] Y.-W. Huang, J. An, H. Huang, X. Li, S.-T. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block partitioning structure in the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 31, no. 10, pp. 3818 - 3833, Otc., 2021.

[4] J. Huo, X. Zhou, H. Yuan, S. Wan, and F. Yang, "Fast Rate-Distortion Optimization for Depth Maps in 3-D Video Coding," *IEEE Trans. Broadcast.,* vol. 69, no. 1, pp. 21 - 32, Jul., 2023.

[5] T. Tang, L. Li, and J. Li, "Improved hierarchical quantisation parameter setting method for screen content coding in high efficiency video coding," *IET Image Process.,* vol. 13, no. 8, pp. 1382-1390, Jun., 2019.

[6] H. Amer, and E.-h. Yang, "Adaptive quantization parameter selection for low-delay HEVC via temporal propagation length estimation," *Signal Process. Image Commun.,* vol. 84, no. 2020, pp. 1-12, Mar., 2020.

[7] Y. Gong, K. Yang, Y. Liu, K.-P. Lim, N. Ling, and H. R. Wu, "Quantization parameter cascading for surveillance video coding considering all inter reference frames," *IEEE Trans. Image Process.,* vol. 30, pp. 5692 - 5707, Jun., 2021.

[8] H. Yin, X. Huang, D. Li, Y. Lu, and Y. Zhou, "Distortion propagation modeling and its applications on frame level quantization control for predictive video coding," *Signal Process. Image Commun.,* vol. 78, pp. 398-408, Jul., 2019.

[9] L. Jia, K. Jia, and X. Fan, "Adaptive Lagrangian multiplier for quantization parameter cascading in HEVC hierarchical coding," *IEEE Signal Process. Lett.,* vol. 27, no. 7, pp. 1220-1224, Jul., 2020.

[10] M. Bichon, J. L. Tanou, M. Ropert, W. Hamidouche, and L. Morin, "Optimal adaptive quantization based on temporal distortion propagation model for HEVC," *IEEE Trans. Image Process.,* vol. 28, no. 11, pp. 5419 - 5434, Nov., 2019.

[11] T. Yang, C. Zhu, X. Fan, and Q. Peng, "Source distortion temporal propagation model for motion compensated video coding optimization," in Int. Conf. Multimedia Expo (ICME), Melbourne, Australia, 2012, pp. 85-90.

[12] Y. Gao, C. Zhu, S. Li, and T. Yang, "Temporally dependent rate-distortion optimization for low-delay hierarchical video coding," *IEEE Trans. Image Process.,* vol. 26, no. 9, pp. 4457-4470, Jun., 2017.

[13] Y. Gao, C. Zhu, S. Li, and T. Yang, "Source distortion temporal propagation analysis for random-access hierarchical video coding optimization," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 29, no. 2, pp. 546 - 559, Feb., 2019.

[14] G. Xu, S. Jin, KaichenTang, Z. Zhang, and Y. Zhou, "LCU-level rate-distortion optimization for versatile video coding," in Int. Symp.Circuits Syst. (ISCAS), Austin, TX, USA, 2022, pp. 3032 - 3027.

[15] Y. Chen, S. Wang, H. Ip, and S. Kwong, "Rate distortion optimization with adaptive content modeling for random-access versatile video coding," *Inf. Sci.,* vol. 645, no. 2023, pp. 1-16, Jnu., 2023.

[16] J. Enhorn, C. Hollmann, R. Sjöberg, J. Ström, and P. Wennersten, "Block importance mapping for video encoding," in Int. Conf. Vis. Commun. Image Process. (VCIP), Suzhou, China, 2022, pp. 1 - 4.

[17] J. Enhorn, C. Hollmann, R. Sjöberg, K. Andersson, and P. Wennersten, "Improvements to a temporal filter for video coding," in Int. Symp.Circuits Syst. (ISCAS), Austin, TX, USA, 2022, pp. 834 - 838.

[18] P. Wennersten, J. Enhorn, C. Hollmann, and J. Ström, *AHG10: Block importance mapping*, Document JVET-Y0077, teleconference, Jan. 2022.

[19] H. Guo, C. Zhu, M. Ye, L. Luo, and X. Yang, "Pre-encoding based temporal dependent rate-distortion optimization for HEVC," *Signal Process. Image Commun.,* vol. 115, no. 2023, pp. 1-15, Mar., 2023.

[20] H. Guo, C. Zhu, J. Chen, L. Luo, Y. Huo, and Y. Liu, "Distortion propagation factor for VVC low-delay hierarchical coding," *IEEE Trans. Broadcast.,* vol. 71, no. 2, pp. 492 - 505, Jun., 2025.

[21] H. Guo, C. Zhu, J. Chen, L. Luo, Y. Huo, and Y. Liu, *Distortion propagation factor for VVC low-delay configuration*, Document JVET-AH0078, Rennes, FR, Apr. 2024.

[22] X. Jin, H. Sun, and Y. Zhang, "Research on VVC intra-frame bit allocation scheme based on significance detection," *Appl. Sci.,* vol. 14, no. 1, pp. 1-19, Jan., 2024.

[23] M. H. Hyun, B. Lee, and M. Kim, "A VVC intra rate control with small bit fluctuations using a Lagrange multiplier adjustment," *IEEE Trans. Multimedia,* vol. 26, pp. 6811 - 6821, Jan., 2024.

[24] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *VTM common test conditions and software reference configurations for SDR video,* document JVET-T2010, teleconference, Oct. 2020.