



HAL
open science

Fast Coding Mode Prediction for Intra Prediction in VVC SCC

Dayong Wang, Junyi Yu, Xin Lu, Frédéric Dufaux, Hongwei Guo, Hui Guo,
Ce Zhu

► **To cite this version:**

Dayong Wang, Junyi Yu, Xin Lu, Frédéric Dufaux, Hongwei Guo, et al.. Fast Coding Mode Prediction for Intra Prediction in VVC SCC. International Conference on Image Processing (ICIP'2024), IEEE, Oct 2024, Abu Dhabi, United Arab Emirates. hal-04609740

HAL Id: hal-04609740

<https://hal.science/hal-04609740v1>

Submitted on 12 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FAST CODING MODE PREDICTION FOR INTRA PREDICTION IN VVC SCC

Dayong Wang^{1,2}, Junyi Yu³, Xin Lu⁴, Frederic Dufaux⁵, Hongwei Guo⁶, Hui Guo^{2*}, Ce Zhu⁷

¹Chongqing Key Laboratory on Big Data for Bio Intelligence,
Chongqing University of Posts and Telecommunications.

²Guangxi Key Laboratory of Machine Vision and Intelligent Control, Wuzhou University.

³Chongqing Key Laboratory of Image Cognition,
Chongqing University of Posts and Telecommunications.

⁴Faculty of Computing, Engineering and Media (CEM), De Montfort University.

⁵Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes.

⁶School of Engineering, Honghe University, Mengzi, Yunnan, China.

⁷University of Electronic Science and Technology of China.

wangdayong@cqupt.edu.cn, yujy95@qq.com, xin.lu@dmu.ac.uk,

frederic.dufaux@12s.centralesupelec.fr, ghw202@foxmail.com, eczhu@uestc.edu.cn

ABSTRACT

Currently, screen content video applications are increasingly widespread in our daily lives. The latest Screen Content Coding (SCC) standard, known as Versatile Video Coding (VVC) SCC, employs screen content Coding Modes (CMs) selection. While VVC SCC achieves high coding efficiency, its coding complexity poses a significant obstacle to the further widespread adoption of screen content video. Hence, it is crucial to enhance the coding speed of VVC SCC. In this paper, we propose a fast mode and splitting decision for Intra prediction in VVC SCC. Specifically, we initially exploit deep learning techniques to predict content types for all CUs. Subsequently, we examine CM distributions of different content types to predict candidate CMs for CUs. We then introduce early skip and early terminate CM decisions for different content types of CUs to further eliminate unlikely CMs. Finally, we develop Block-based Differential Pulse-Code Modulation (BDPCM) early termination to improve coding speed. Experimental results demonstrate that the proposed algorithm can improve coding speed by 34.95% on average while maintaining almost the same coding efficiency.

Index Terms— VVC SCC, content type, fast coding mode decision, BDPCM.

*Corresponding author

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62020106011, Grant 62162054, Grant 61961036, and Grant 62061015; in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZDK202100604; in part by the Guizhou Provincial Basic Research Program (Natural Science) Qiankehejichu-ZK[2023]Yiban252; and in part by the Guangxi Key Laboratory of Machine Vision and Intelligent Control under Grant 2023B05.

1. INTRODUCTION

Content video refers to recordings captured directly from the display screen of an electronic device. The surge in popularity of computer screen-sharing applications can be attributed to rapid advancements in networking and portable devices in recent times. Such applications, including remote desktop access and video conferencing, are often complemented by document and slideshow sharing. These multimedia communication sessions typically comprise a blend of Camera-captured Content (CC) and Screen Content (SC). Additionally, numerous television programs and online videos also integrate both CC and SC elements. It is foreseeable that the future will witness the emergence of cloud services utilizing screen-sharing technologies [1]. Consequently, there is a growing demand for the efficient compression of SC. In response to this demand, a Call for Proposal (CfP) was issued in January 2014 [2] for the development of Screen Content Coding (SCC) [3] as an extension of the High Efficiency Video Coding (HEVC) standard [4]. The SCC extension introduced two major Coding Modes (CMs): Intra Block Copy (IBC) [5] and Palette (PLT) [6] modes, aimed at improving coding efficiency. In recent times, the Joint Video Experts Team (JVET) has spearheaded the development of the next-generation video coding standard, Versatile Video Coding (VVC) [7]. Early in its development, VVC incorporated IBC, and later, PLT was added in subsequent phases. However, the coding process of VVC is extremely complex. Consequently, the development of fast algorithms to mitigate the coding complexity of VVC SCC is highly desirable.

This paper introduces a fast Intra prediction algorithm aimed at enhancing the coding speed of VVC SCC. Our

approach first leverages the deep learning techniques to categorize CUs into distinct types, namely Animation (A) CUs, Camera-captured Content (CC) CUs, Text and Graphics with Motion (TGM) CUs, and Mixed content (Mixed) CUs. Based on this categorization, we predict candidate CMs and subsequently implement Block-based Differential Pulse-Code Modulation (BDPCM) early termination to improve the coding speed.

2. RELATED WORK

In general, CM prediction can be broadly categorized into two groups: content classification-based methods and feature classification-based methods. A detailed review of these methods is presented below.

Content classification-based methods: Due to the relationship between image content and CMs, candidate CMs can be derived based on their content. In [8], statistical features were selected and fed into a decision tree to effectively classify CUs into two categories: natural content CUs and screen content CUs, aiming to predict candidate CMs. In [9], factors such as color number and gradient were utilized to predict content types of CUs, thereby predicting candidate CMs. Gao et al. [10] employed a CNN to classify CUs into natural, text, image, and color content blocks. They then integrated CU content type, CU size, mode complexity, and spatial correlations between neighboring CUs and the current CU to predict candidate CMs. These methods predict candidate CMs based on content types. However, since the correlations between content types and CMs are not consistently robust, relying on content to predict candidate CMs may not always yield optimal performance.

Feature classification-based methods: Tsang et al. [11] predicted CU sizes by considering the RD costs and early skipped the IBC mode based on the hash value of each block. Kuang et al. [12] employed an online learning Bayesian decision rule to skip unlikely CMs early. Additionally, Kuang et al. [13] first designed various features as eigenvalues and then employed a decision tree to sequentially determine whether to examine Intra mode, IBC Merge mode, IBC search mode, and PLT mode. Tsang et al. [14] selected a more extensive set of statistical features to construct a random forest to determine whether to examine Intra mode, IBC Merge mode, IBC Search mode, and PLT mode. Kuang et al. [15] first chose statistical features and then used an online-learning-based Bayesian decision rule to determine whether to early skip unlikely CMs. Subsequently, they utilized the mode information from neighboring CUs to further eliminate unlikely CMs for the current CU. Kuang et al. [16] developed a fast prediction network based on deep learning to obtain probabilities for all CMs and to early skip CMs of low probabilities, thus improving coding speed. Tsang et al. [17] used a CNN-based classifier to predict candidate CMs for all 4×4 CUs and then determined the CMs based on the counterparts of their

4×4 sub-CUs. The aforementioned methods predict candidate CMs based on features. However, as the correlations between features and CMs are not consistently robust, utilizing features for predicting candidate CMs may not always yield optimal performance.

3. DEEP LEARNING-BASED CONTENT CLASSIFICATION (DLBCC)

SCC sequences comprise four types of content: A, CC, TGM, and Mixed. Typically, each content type exhibits a strong correlation with specific CMs and CU splitting that are most likely to be selected for coding. Consequently, the first step of the proposed algorithm involves categorizing CU contents, followed by predicting candidate CMs based on their corresponding CU contents. To categorize CU contents, we designed a neural network and subsequently trained its associated model.

3.1. Design of the Neural Network

In VVC, a CTU has a default size of 128×128 and can be flexibly partitioned into blocks with a minimum size of 4×4 . The coding structure of QTMT allows the partitioning of a CTU into CUs of up to 17 different sizes. Employing deep learning techniques directly to predict candidate CMs for all CUs makes the computational process unbearably complex. Given that the minimum size of CUs is 4×4 , we can first determine the content types of 4×4 CUs and subsequently predict the content type of each CU based on its 4×4 sub-CUs in the lower partition layer. If all 4×4 sub-CUs within a CU exhibit the same type, the CU is categorized as that specific type. Conversely, if the 4×4 sub-CUs within a CU differ in type, the CU is classified as a Mixed type. Determining the type of each 4×4 CU is a challenging task.

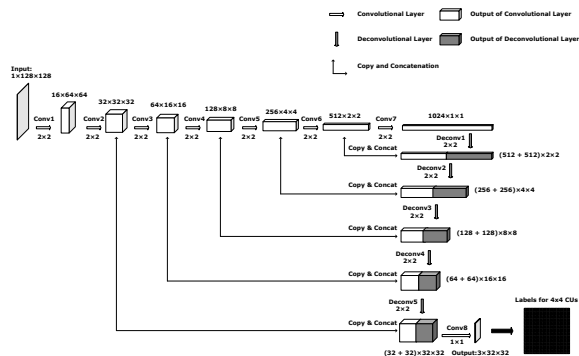


Fig. 1. Structure of Sub-CUNet.

Deep learning has demonstrated the capability to automatically extract extensive features, yielding exceptional performance across various applications. The Fully Convolutional Network (FCN) has proven to be highly effective,

particularly for various pixel-wise segmentation tasks such as semantic segmentation, instance segmentation, and biomedical image segmentation. Among these FCNs, the U-Net stands out as a well-known choice for biomedical image segmentation [16] [17]. Inspired by U-Net, we have developed our own 4×4 sub-CU segmentation net, denoted as Sub-CUNet [17]. Unlike traditional pixel-wise segmentation, our Sub-CUNet produces only 4×4 sub-CU-wise labels, as illustrated in Fig. 1.

The luminance component of a 128×128 CTU serves as the input to Sub-CUNet. Sub-CUNet generates 1024 labels corresponding to 1024 4×4 sub-CUs, with each label representing the content most likely to be chosen. Sub-CUNet is composed of eight convolutional layers (conv1-conv8), five deconvolutional layers (deconv1-deconv5), and five concatenation layers (concat1-concat5). Following each convolutional or deconvolutional layer, a Rectified Linear Unit (ReLU) activation function is applied. However, in the case of conv8, the softmax activation function is employed to generate the output labels.

Through the Sub-CUNet, a 128×128 CTU generates 1024 labels, each corresponding to a 4×4 sub-block. Each 4×4 sub-block is categorized into one of the four content types according to the probability. If all 4×4 sub-blocks within a CU share the same content type, the CU is classified as that specific content type. Otherwise, the current CU is classified as Mixed.

3.2. Training of the Neural Network

To avoid any overlap between the training set and the test set, we selected training sequences that are not included in the Common Test Conditions (CTC) [18] to generate training samples. These sequences cover A, CC, and TGM content, specifically including ClearTypeSpreadsheet, KristenAndSaraScreen, MissionControlClip1, ParkScene, PcbLayout, PeopleInVehicle, PptDocXls, RealTimeData, Seeking, VideoConferencingDocSharing, WordEditing, DOTA2, CSGO, WITCHER3, GTAV, and Hearthstone. In addition to those sequences recommended in [19], we also obtained some sequences from the internet.

To fully leverage computing resources, the batch size is set to 4096. Sequences are randomly allocated at a ratio of 8:2 to the training set and the validation set. After loading the dataset, pixel values are normalized. In the training stage, PyTorch is utilized on an RTX3090 GPU. The cross-entropy loss function is selected, and the optimizer used is Adam, with default values of 0.9 for momentum 1 and 0.999 for momentum 2. The learning rate is set to 0.0001, and the epoch is set to 1000. The loss during the training process is illustrated in Fig. 2, and the best model is determined based on the decreasing trend of the loss value during training.

From Fig. 2, it is evident that as the number of epochs increases, both training and validation loss values gradually

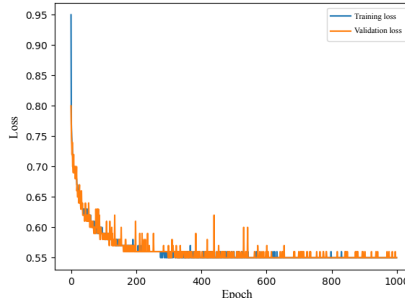


Fig. 2. Loss during the training process of the content classifier.

decrease and converge to similar values. When the number of epochs is greater than or equal to 800, both loss values reach a minimum and maintain stability with some slight fluctuations. This observation indicates that the networks are not experiencing overfitting problems, which typically occur when training losses are very small but validation losses are very high.

4. CONTENT AND FEATURE-BASED CM PREDICTION (CFBCMP)

Through DLBCC, we can obtain information about the types of CUs. Subsequently, we examine CM distributions for CUs of various types and then identify likely CMs while excluding unlikely ones based on the obtained distributions.

To obtain the CM distribution for CUs of various types, extensive experiments were conducted. In our experiments, one sequence from each type was selected to cover different types. Specifically, the test sequences outlined in [16] were utilized. According to the CTC, all sequences were encoded using QP values of 22, 27, 32, and 37 under the All_Intra configuration. The corresponding CM distribution for CUs of various types is detailed in Table 1.

Table 1. CM distributions for CUs of various content types

Categories	Intra (%)	PLT (%)	IBC (%)
A	93.14	1.51	5.34
CC	98.24	0.15	1.61
Mixed	59.1	12.6	28.3
TGM	33.38	26.37	40.26

From Table 1, it is evident that the CM distribution for CUs of various types exhibits significant differences. Specifically, CUs in CC predominantly employ the Intra mode. Consequently, CUs in CC exclusively examine the Intra mode. Given the small proportion of CUs in Mixed, we directly examine all three CMs. Regarding CUs in A, while the Intra mode has a high proportion, the proportions of the other two CMs are not considered negligible. As for CUs in TGM, the

differences in proportions among all three CMs are not substantial. Therefore, for CUs in either A or TGM, it is not feasible to directly predict the candidate CMs based on their CM distributions.

Considering both prediction complexity and accuracy, we chose Decision Tree (DT) as the classifier in this paper. DT is a widely used binary classification approach known for its comprehensible flow-chart-like structure. The DT is constructed using "Scikit-learn" in Python with offline training. The aforementioned sequences used for DLBCC are selected as training sequences.

As mentioned above, we only need to predict candidate CMs for CUs in A and TGM. As shown in Table 1, for CUs in both A and TGM, the Intra mode is frequently used. Moreover, these CMs are strongly related to textural features and can be directly predicted based on them. Therefore, it is advisable to first check the Intra mode. Since the coding process of the PLT mode is significantly simpler than that of the IBC mode, which involves IBC merge and IBC search, we next check the PLT mode and finally the IBC mode.

For the convenience of subsequent descriptions, features obtained directly before encoding are denoted as Direct Features (DFs), and features obtained during encoding are denoted as Coding Features (CFs). To enhance coding speed, before examining a CM, we can use some DFs to determine whether the CM can be skipped early. If affirmative, the CM can be directly skipped. Otherwise, we examine it and combine DFs with CFs to determine whether the CM is the best one, thus early terminating the CM selection.

4.1. Mode Early Skipping based on DFs

Before examining a CM, certain DFs can be acquired. We input these DFs into a DT to determine the confidence level of utilizing the CM. To accurately obtain the confidence level, the DFs are selected as follows.

A CTU can be partitioned into CUs of up to 17 different sizes using the QTMT structure. The CM distribution for CUs of different sizes is significantly different. Clearly, the choice of the best CM is closely tied to the size of a CU. As a result, the size of CUs is chosen as a feature.

In general, the texture of a CU plays a significant role in the decision-making process for CM selection. Pixel variances are evidently crucial textural features in this context. Consequently, we utilize overall variances (OVP), as well as horizontal (HVP) and vertical (VVP) variances of pixels within a CU as features.

The Number of Background Color (NBC) is a metric representing the quantity of pixels that share the same color, with the highest frequency of occurrence within a given CU. The Number of Distinct Color (NDC) measures the pixels in a CU with unique pixel values, essentially pixels with colors distinct from others. Studies in [13] [14] demonstrate that the CM selection of PLT is strongly related to the count of colors.

This is caused by the fact that the count determines the number of palette items that must be encoded by PLT following clustering. After clustering, the pixel values of the block are represented as indices in the palette, and these indices are encoded using Run-Length Coding. Only one item is required for consecutive identical colors, referred to as a color segment in this context. Consequently, the minimum Number of Color Segments (NCS) observed along both horizontal and vertical scanning paths directly influences the number of index items that PLT needs to encode and significantly impacts the resulting PLT bit rate.

Feeding the features of a CM-to-examine into a DT enables us to derive a confidence level for this CM. If the confidence level is less than or equal to a specified threshold, we can terminate CM selection early. To determine the optimal threshold, we used VTM-17.0 for testing on a server equipped with an Intel Xeon Gold 5122 CPU and 64GB of memory. The aforementioned training sequences are encoded under the All_Intra configuration using QP values of 22, 27, 32, and 37. Coding efficiency is measured using BDBR [20]. The confidence levels and their respective BDBRs for early skipping of all three CMs in the TGM category are illustrated in Fig. 3.

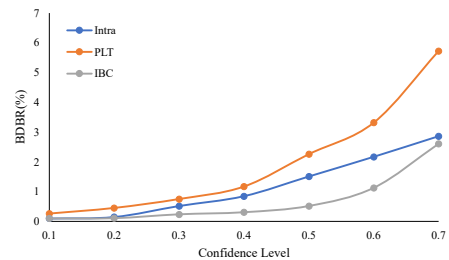


Fig. 3. Confidence level and the respective BDBR for CUs in the TGM category.

From Fig. 3, it is evident that when the confidence levels of both Intra mode and IBC mode are less than or equal to 0.2, the corresponding BDBR remains nearly the same. However, when the confidence levels of both these two CMs exceed 0.2, the corresponding BDBR dramatically increases. Clearly, there is a turning point at a confidence level of 0.2. Therefore, we set the threshold as 0.2 for both Intra mode and IBC mode. As for the PLT mode, when its confidence level is greater than 0.1, the corresponding BDBR continuously increases. Consequently, we set the threshold as 0.1 for this CM. Employing a similar approach, we can determine thresholds for all CMs for CUs in the A category. Through the aforementioned process, we derive the thresholds for all CMs for CUs in the TGM and A categories, which are listed in Table 2.

When the confidence level of a CM for CUs in the TGM and A categories is less than or equal to the corresponding threshold in Table 2, this CM can be skipped directly, thereby improving the coding speed.

Table 2. Thresholds for all CMs for CUs in the TGM and A categories

Categories	Intra	PLT	IBC
A	0.2	0.5	0.2
TGM	0.2	0.1	0.2

4.2. Early Termination Based on DFs and CFs

As mentioned earlier, if the confidence level of a CM exceeds the corresponding threshold in Table 2, further examination of the CM becomes necessary. After examining the CM, specific coding information, including residual coefficients and RD costs, is obtained. This coding information plays a crucial role in the CM early termination. If the residual coefficients and RD costs are effectively insignificant, it is highly likely that the corresponding CM is the best one. Consequently, CM selection can be terminated early. Therefore, considering both textural features and coding information, we can determine whether the mode-to-examine is the best one, thus early terminating the CM selection.

After examining a CM, the acquired textual features and coding information are inputted into a DT to determine the corresponding confidence level. To achieve the optimal confidence level for early termination of the CM, we apply the aforementioned experimental conditions during testing. The experiments demonstrate that, for all three CMs, as the confidence level increases, BDBR significantly decreases. When the confidence level is greater than or equal to 0.9, BDBR remains nearly constant. Clearly, there is a turning point when the confidence level equals 0.9. Therefore, we set the threshold as 0.9 for the early termination of all three CMs.

5. CONTENT AND RESIDUAL COEFFICIENT-BASED BDPCM EARLY TERMINATION (CRCBBET)

In VVC, the Intra prediction process primarily involves three steps: Rough Mode Decision (RMD), BDPCM, and Rate-Distortion Optimization (RDO). RMD uses the Hadamard transform to establish a Rate-Distortion Mode List (RDL). BDPCM employs only horizontal and vertical modes for prediction, directly quantifying their residual coefficients without using a transformation. Subsequently, the RDO process uses the DCT transform to examine the RDL, Matrix Weighted Intra Prediction (MIP), and Intra Sub-Partition (ISP) to determine the best mode. Given the nature of BDPCM, as mentioned above, its coding complexity is clearly insignificant. Once BDPCM has been examined, if it is identified as the best mode, subsequent coding processes can be early terminated, thereby significantly improving coding speed. To determine whether BDPCM is the best mode, we examine both the percentage of cases in which BDPCM is selected as the best mode and the percentage of cases in which

RDO determines the best mode for CUs in the TGM and A categories under the aforementioned experimental conditions. The results are shown in Table 3.

Table 3. Comparison of BDPCM and RDO for CUs in TGM and A categories

Categories	BDPCM(%)	RDO (%)
A	2.50	97.50
TGM	38.85	61.15

Table 3 shows the proportions of BDPCM in the A and TGM categories, which are 2.50% and 38.85%, respectively. Given the relatively small proportion of BDPCM in the A category, our focus shifts to determining whether BDPCM is the best mode for CUs in the TGM category.

The performance of BDPCM can be effectively reflected by the residual coefficients. When examining a CU using BDPCM, if the values of its residual coefficients are small and evenly distributed, this indicates that the CU is predicted very well. BDPCM is thus highly likely to be the best mode, and the subsequent RDO processes can be terminated early. As the variance of residual coefficients can accurately measure their complexity, we select the overall variance (OVRB), horizontal variance (HVRB), and vertical variance (VVRB) of residual coefficients as features. Additionally, if the proportion of zero values in the quantized residual coefficients is notably high, it also indicates that the CU is predicted very well, making BDPCM likely to be the best mode. Therefore, the proportion of zero values in the quantized residual coefficients, denoted as PZQR, is also chosen as a feature.

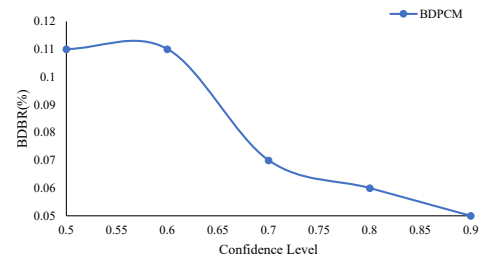


Fig. 4. Thresholds and their respective BDBR for CUs in the TGM category.

Feeding these features into a DT allows us to determine the confidence level of utilizing BDPCM as the best mode. Clearly, confidence levels are closely linked to BDPCM selection. If the confidence level is greater than or equal to a predefined threshold, BDPCM can be chosen as the best mode, leading to the early termination of subsequent coding processes. To identify the optimal threshold, the aforementioned experimental conditions and sequences are employed in testing. Theoretically, only when thresholds are greater than or equal to 0.5 can BDPCM be selected as the best mode. Therefore, we exclusively use thresholds greater than or equal

Table 4. Overall performance comparison among different algorithms

Categories	Sequences	Proposed		MLMD [13]		OLBD [15]		FSCNET [17]	
		BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)	BDBR (%)	ΔT (%)
A	Robot	0.92	22.17	1.57	17.49	0.69	18.50	1.98	37.33
CC	EBURainFruits	0.01	31.07	1.32	16.87	0.27	28.01	0.07	36.44
	Kimono1	1.07	29.12	2.13	16.77	1.79	38.37	-0.07	24.66
Mixed	BasketballScreen	1.00	35.52	1.44	21.90	0.72	10.01	4.38	25.16
	MissionControlClip2	0.67	34.36	1.22	22.19	0.67	10.98	2.17	31.23
	MissionControlClip3	0.57	37.37	1.10	24.59	0.47	9.17	5.71	29.32
TGM	Console	0.43	47.14	2.33	27.37	0.49	16.74	1.98	23.69
	Desktop	0.10	59.65	1.56	30.89	0.5	20.72	2.24	32.29
	FlyingGraphics	3.11	39.49	1.71	24.28	0.57	13.19	6.66	18.51
	Map	1.18	18.60	1.65	20.56	1.03	10.84	2.28	26.87
	Programming	1.17	34.69	1.29	23.09	0.79	9.73	3.64	24.55
	SlideShow	0.71	15.55	1.74	17.86	0.85	11.01	3.77	23.19
	WebBrowsing	0.39	49.66	1.87	26.35	0.46	17.35	7.01	17.62
All	Average	0.88	34.95	1.61	22.32	0.72	16.51	3.22	26.98

to 0.5 for testing purposes. The corresponding thresholds and their associated BDBRs are illustrated in Fig. 4.

From Fig. 4, it is evident that as the thresholds increase, the corresponding BDBR gradually decreases. When a CU is examined by BDPCM, if its confidence level is greater than or equal to 0.5, BDPCM is selected as the best mode, and subsequent RDO processes can be terminated early.

6. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, we employed the VTM-17.0 reference software for testing on a server equipped with an Intel Xeon Gold 5122 CPU and 64GB of memory. The experimental parameters were configured in accordance with the CTC. Specifically, both GOPSize and IntraPeriod were set to 1, and QPs were set as (22, 27, 32, 37). The coding efficiency was evaluated using BDBR, representing the percentage of bitrate increase for the same quality compared to the reference software. Positive and negative values indicate a loss and an improvement in coding efficiency, respectively. Computational complexity was measured by the percentage of the coding time savings, denoted as ΔT .

Considering the limited research on VVC SCC and the abundance of works on HEVC SCC, we compare our overall performance with MLMD [13], and OLBD [15], FSCNET [17]. It is important to note that we did not compare with the latest two papers, namely Gao et al.’s algorithm [10] and Zuo et al.’s algorithm [21]. The rationale behind this decision is that Gao et al.’s algorithm assigns different CMs to CUs of various sizes in HEVC, which encompasses only four CUs of different sizes. In contrast, VVC supports CUs of 17 different sizes. Consequently, Gao’s method is not suitable for a direct comparison with our proposed algorithm. Incorporating Gao et al.’s algorithm into VVC would result in a very limited time reduction, rendering it an unfair basis for comparison. The coding speed of Zuo et al.’s algorithm has increased by 90.51%, but this improvement comes at the cost of a 9.47% increase in BDBR. Despite the notable improvement in coding speed, the coding efficiency of Zuo et al.’s

algorithm has significantly deteriorated. This degradation in coding efficiency cannot be deemed an acceptable trade-off for improved coding performance. Consequently, Zuo et al.’s algorithm is not an ideal benchmark for comparison. To the best of our knowledge, these three algorithms are the state-of-the-art fast Intra coding algorithms for VVC SCC. To ensure a fair comparison, we implement these algorithms on the same computing platform and use identical experimental parameters. The comparison of coding performance and coding speed is presented in Table 4. In consideration of the practicality of our algorithm, the inference time of the models employed in our algorithm has been included in the experimental results.

From Table 4, it can be observed that coding speed improvements achieved by our proposed algorithms, MLMD, OLBD, and FSCNET, are 34.95%, 22.32%, 16.51% and 26.98%, respectively. The corresponding average BDBR increases by 0.88%, 1.61%, 0.72%, and 3.22%, respectively. When compared with FSCNET and MLMD, our proposed algorithm performs significantly better in terms of both coding speed and coding efficiency. In comparison to OLBD, the proposed algorithm is faster by 18.44% with a 0.16% BDBR increase. Although the proposed algorithm exhibits slight coding efficiency losses, they can be considered negligible when considering the substantial coding speed improvements.

7. CONCLUSIONS

In this paper, we propose a fast CM prediction for Intra prediction in VVC SCC. Experimental results demonstrate that the proposed algorithm can significantly enhance coding speed with negligible impact on coding efficiency. In our future work, we plan to develop efficient deep learning-based algorithms to further improve the coding speed of VVC SCC.

8. REFERENCES

- [1] Y. Lu, S. Li, and H. Shen, “Virtualized screen: A third element for cloudmobile convergence,” *IEEE MultiMe-*

- dia*, vol. 18, no. 2, pp. 4–11, 2011.
- [2] “Joint call for proposals for coding of screen content,” *ISO/IEC JTC1/SC29/WG11*, N14175, San Jose, CA, USA, Jan. 2014.
- [3] J. Xu, R. Joshi, and R. A. Cohen, “Overview of the emerging HEVC screen content coding extension,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 50–62, 2016.
- [4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [5] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, K. Rapaka, C. Pang, V. Seregin, Y.-K. Wang, and M. Karczewicz, “Intra block copy in HEVC screen content coding extensions,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409–419, 2016.
- [6] Z. Ma, W. Wang, M. Xu, and H. Yu, “Advanced screen content coding using color table and index map,” *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4399–4412, 2014.
- [7] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, “Overview of the versatile video coding (VVC) standard and its applications,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.
- [8] H. Yang, L. Shen, and P. An, “An efficient intra coding algorithm based on statistical learning for screen content coding,” in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2468–2472.
- [9] J. Lei, D. Li, Z. Pan, Z. Sun, S. Kwong, and C. Hou, “Fast intra prediction based on content property analysis for low complexity HEVC-based screen content coding,” *IEEE Transactions on Broadcasting*, vol. 63, no. 1, pp. 48–58, 2017.
- [10] C. Gao, L. Li, D. Liu, Z. Chen, W. Li, and F. Wu, “Two-step fast mode decision for intra coding of screen content,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 5608–5622, 2022.
- [11] S.-H. Tsang, Y.-L. Chan, W. Kuang, and W.-C. Siu, “Reduced-complexity intra block copy (IntraBC) mode with early cu splitting and pruning for HEVC screen content coding,” *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 269–283, 2019.
- [12] W. Kuang, S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, “Fast mode decision algorithm for HEVC screen content intra coding,” in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2473–2477.
- [13] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “Machine learning-based fast intra mode decision for HEVC screen content coding via decision trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1481–1496, 2020.
- [14] S.-H. Tsang, Y.-L. Chan, and W. Kuang, “Mode skipping for HEVC screen content coding via random forest,” *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2433–2446, 2019.
- [15] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “Online-learning-based bayesian decision rule for fast intra mode and cu partitioning algorithm in HEVC screen content coding,” *IEEE Transactions on Image Processing*, vol. 29, pp. 170–185, 2020.
- [16] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, “DeepSCC: Deep learning-based fast prediction network for screen content coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 1917–1932, 2020.
- [17] S.-H. Tsang, N.-W. Kwong, and Y.-L. Chan, “FastSCC-Net: Fast mode decision in VVC screen content coding via fully convolutional network,” in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 2020, pp. 177–180.
- [18] C. Yung-Hsuan, S. Yu-Chen, X. Jizheng, and X. Xiaozhong, “JVET common test conditions and software reference configurations for non-4:2:0 colour formats,” document JVET, JVET-R2013, Teleconference, Apr. 2020.
- [19] K. S. Karl Sharman, “Common test conditions,” document JCTVC-X1100, 26th Meeting: Geneva, CH, January 2017.
- [20] G. Bjontegaard, “Calculation of average PSNR differences between RD-Curves,” *ITU T Q/sg16 Vceg Vceg M*, 2001.
- [21] Y. Zuo, C. Gao, D. Liu, L. Li, Y. Zhang, and X. Sun, “Learned rate-distortion cost prediction for ultrafast screen content intra coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023.