Efficient Mode Decision for HEVC Screen Content Coding by Content Analysis

Wei Kuang, Yui-Lam Chan, Sik-Ho Tsang
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
wei.kuang@connect.polyu.hk, enylchan@polyu.edu.hk, sik-ho.tsang@polyu.edu.hk,

***Abstract*—The screen content coding (SCC) extension of High Efficiency Video Coding adopts three modes, the conventional intra mode of HEVC, the new intra block copy mode and palette mode, to improve the coding performance of screen content videos. However, the exhaustive search for the optimal mode among the three mode candidates brings significant computational burden to a SCC encoder. This paper proposes an efficient mode decision algorithm for SCC by content analysis. Since screen content videos contain both stationary coding units (CUs) and dynamic CUs, two different techniques are proposed. For dynamic regions, the high gradient pixels and the background color are jointly analyzed to perform CU type classification, and then different mode candidates are checked adaptively according to the CU type. For stationary CUs, the information from the collocated CUs is utilized to predict the optimal mode. Experimental results show that the proposed algorithm achieves 40.05% computational complexity reduction on average with 1.40% Bjøntegaard delta bitrate loss under All Intra configuration.**

Keywords—screen content coding (SCC), High Efficiency Video Coding (HEVC), mode decision, fast algorithm.

I. Introduction

Screen content videos refer to videos captured from the display screens of electronic devices, and they play an important role in many screen sharing based applications, such as cloud-mobile computing, remote education, video conference with document sharing, wireless or Wi-Fi screen mirroring [1]. Screen content videos usually show mixed content of the traditional natural image blocks (NIBs) and the new computer-generated screen content blocks (SCBs), as shown in Fig. 1. To explore new coding schemes for screen content videos, the Joint Collaborative Team on Video Coding (JCT-VC) started a Screen Content Coding (SCC) extension [2] to High Efficiency Video Coding (HEVC) [3] in 2014.

Since NIBs can be efficiently encoded by the conventional intra (Intra) mode of HEVC, screen content coding (SCC) directly inherits Intra mode from HEVC. However, SCBs show very different characteristics from NIBs, and it makes Intra mode inefficient when encoding SCBs. A SCB usually has many high gradient pixels, limited colors, many repeated patterns within one frame, and a large area filled with a background color. Besides, many SCBs are stationary blocks, which have the same content as their collocated blocks. To improve the coding efficiency of SCBs, two new modes, intra block copy (IBC) [4] and palette (PLT) [5, 6] modes are specially designed in SCC. As a result, the exhaustive search for optimal mode among the three mode candidates leads to great computational burden to a SCC encoder, and the urgent requirement for fast SCC encoding algorithm is on the rise.

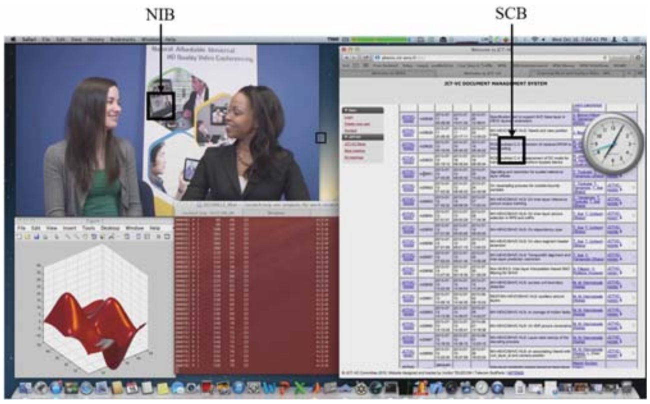

Fig. 1. A frame in a screen content sequence “MissionControlClip3”.

In the literature, existing fast SCC encoding algorithms can be divided into three categories. The first is to simplify mode decision [7-11]. Since IBC is found to have the highest computational complexity among Intra, IBC and PLT modes, [7, 8] were proposed to speed up the IBC mode, where hash values, CU activities, and CU gradients are calculated to adaptively check IBC mode. In [9], Bayesian decision rules are utilized by using online learning to eliminate the unnecessary mode candidates. In [10, 11], flexible mode classifiers were proposed by decision trees and random forests.

The second category is to early predict the CU partitioning decision [12, 13]. In [12], neural network-based classifiers were trained by extracting features describing CU statistics and sub-CU homogeneity, and early CU partitioning decision is made according to the outcome of those classifiers. However, it induces high rate distortion (RD) performance loss because of low prediction accuracy. In [13], the CU partitioning structure of the collocated CU is used to predict the depth level of the current CU if they have similar content. However, it is inefficient for sequences with many dynamic contents, and it needs to disable the fast algorithm every ten frames to avoid error propagation.

In the third category, approaches were proposed to make both fast mode decision and fast CU partitioning decision [14-19]. In [14, 15], content dependent rules were proposed to skip unnecessary modes and CUs by investigating learning frames. However, they cannot simplify the encoding process of the learning frames. Decision tree-based classifiers are utilized in [16, 17] with off-line training, where CUs are classified into

This work was supported by The Hong Kong Polytechnic University Internal Research Grant (ZZHR).

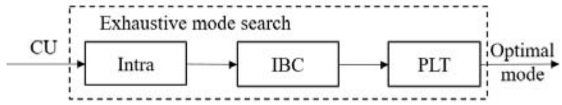

Fig. 2. Exhaustive mode search in SCC.

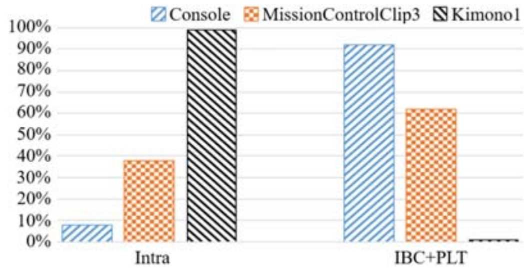

Fig. 3. Optimal mode distributions with $d = 2$ and QP of 32.

SCBs and NIBs by analyzing various features. SCBs only check IBC and PLT modes, while NIBs only check Intra mode. Then, classifiers for making CU partitioning decision were also proposed. However, Intra mode is always checked for all CUs with 2N×2N prediction units (PUs) to get features required by the classifiers in [17]. Therefore, it induces high computational overhead. In [18], CUs are also classified into NIBs and SCBs by CU content analysis. Then, the depth levels from the collocated CU and spatial neighboring CUs, as well as the coding bits of the current CU are analyzed to make early CU partitioning decision. In [19], IBC mode are adaptively skipped by checking a hash value, and early CU partitioning decision is also made by checking RD cost. In [20], a deep-learning-based fast approach was proposed but it induces relatively high computational overhead.

In this paper, we propose an efficient mode decision algorithm by content analysis. For a dynamic CU with different content from its collocated CU, a simple yet efficient CU type classification approach is proposed to classify it as a SCB or a NIB, and it is achieved by analyzing the high gradient pixels and the background color in a CU. Then, NIBs only check Intra mode while SCBs only check IBC and PLT modes. For a stationary CU which has the same content as its collocated CU, the optimal mode of the collocated CU is utilized to predict the optimal mode of the current CU, and then the incorrect prediction is eliminated by comparing RD cost. The differences between our contributions and the related schemes can be summarized as: 1) Unlike [7, 8] simplifying IBC mode and [12, 13] simplifying CU partitioning decision, the proposed algorithm reduces unnecessary modes in each CU, so that they can be integrated together to provide higher encoding time reduction. 2). The fast mode decision approaches in [9-11, 14-19] does not utilize the temporal correlation of the stationary CUs as our proposed algorithm. Besides, the proposed CU type classification for dynamic CUs require less features, so that it induces less computational overhead than the machine-leaning/deep-learning-based approaches [10, 11, 16, 17, 20].

The rest of this paper is organized as follows. Section II presents the review and analysis of the mode decision in SCC. Section III presents the proposed efficient mode decision algorithm for SCC. The experimental results are presented in Section IV to verify the performance of the proposed algorithm. Finally, section V concludes the paper.

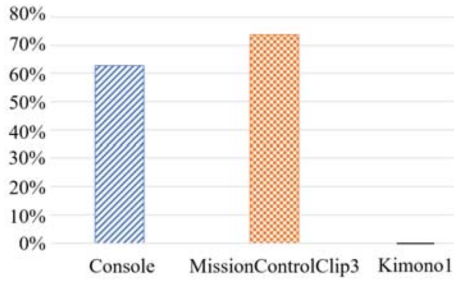

Fig. 4. The percentage of stationary CUs.

II. Review and Analysis of the Mode Decision in SCC

SCC inherits the same flexible quad-tree block partitioning structure from HEVC. A coding tree unit (CTU) of 64×64 pixels (depth level $d = 0$) can be partitioned to four CUs of equal size, and then each CU can be further partitioned into four smaller CUs recursively until the smallest CUs of 8 × 8 pixels (depth level $d = 3$) are reached. For each CU, all modes are exhaustively searched, as shown in Fig. 2, and the one with the small RD cost is selected as the optimal mode.

Intra mode is inherited from HEVC to SCC, and it is designed based on the assumption that the content in a NIB has a dominant direction, and a NIB is predicted by its boundary pixels along the dominant direction. IBC mode is designed based on the observation that there are many repeated patterns for SCBs in the current frame. IBC mode is a block matching-based approach, which performs motion estimation in the reconstructed areas of the current frame, and matching blocks are used to predict the current CU. PLT mode is designed based on the observation that a SCB contains limited number of colors. PLT mode utilizes a palette table and an index map to encode a CU. The palette table is predicted from the previously coded CUs, and it contains several representative samples. Then the index map denotes the position of the representative samples in the CU.

To analyze the optimal mode distribution, three representative sequences are selected, which are "Kimono1", "Console", and "MissionControlClip3". It should be noted that "Kimono1" is a camera-captured sequence and it only contains NIBs, "Console" only contains SCBs, and "MissionControlClip3" contains mixed content of NIBs and SCBs. The optimal mode distribution with $d = 2$ and quantization parameter (QP) of 32 are shown in Fig. 3. It is observed that almost all CUs in "Console" select IBC and PLT modes, and almost all CUs in "Kimono1" select Intra mode. Besides, all modes take large percentages in "MissionControlClip3" due to the mixed content of NIBs and SCBs. Therefore, if incoming CUs can be classified into NIBs and SCBs with high precision, unnecessary mode candidates can be early skipped to reduce encoding time.

Furthermore, there exist many stationary CUs in a screen content sequence, i.e., the sum of absolute differences (SAD) between the current CU and its collocated CU is 0. Fig. 4 presents the percentage of the stationary CUs for the three representative sequences. It is observed that "Kimono1" does not contain stationary CUs because it is a camera-captured sequence. However, stationary CUs take up high percentages in both "Console" and "MissionControlClip3". If the optimal mode correlation can be investigated and utilized for the stationary CUs, unnecessary mode candidates can be further reduced.

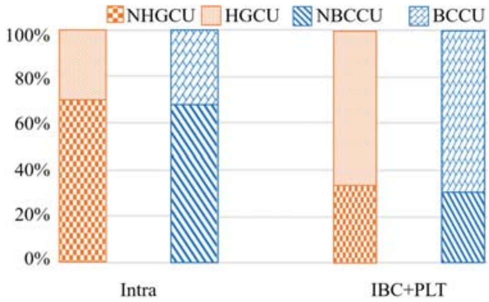

Fig. 5. The distributions of HGCUs, NHGCUs, and BCCUs, NBCCUs over different modes with $d = 2$ and QP of 32.

TABLE I. AVERAGE HIT RATE OF THE PROPOSED EFFICIENT MODE DECISION TECHNIQUE FOR DYNAMIC CUS

Hit Rate	***d* = 0 (%)**	***d* = 1 (%)**	***d* = 2 (%)**	***d* = 3 (%)**
Average	90.78	94.59	95.02	94.63

III. PROPOSED EFFICIENT MODE DECISION

The incoming CUs are divided into stationary CUs and dynamic CUs by comparing their pixel values with the collocated CUs. Then different efficient mode decision techniques are proposed for them separately.

A. Efficient Mode Decision for Dynamic CUs

For dynamic CUs, they are classified into NIBs and SCBs, and then unnecessary modes are eliminated based on the analysis in Section II. Since a SCB usually contains many high gradient pixels and has a large area with background color, these two features are selected to classify SCBs from NIBs.

A high gradient pixel refers to a pixel with a component $C_{i,j}$ ($C \in$ {Y, U, V}) very different from its neighboring pixels $C_{i\pm1,j}$, $C_{i,j\pm1}$. A pixel is detected as a high gradient pixel if

$$|C_{i,j} - C_{i\pm1,j}| > TH\_P_{DIFF} \text{ or } |C_{i,j} - C_{i,j\pm1}| > TH\_P_{DIFF} \quad (1)$$

where i and j denote the row and column indices of a pixel, respectively. TH_P_{DIFF} is a threshold controlling the component difference between neighboring pixels, which is set to 32 empirically. Thus, if the component difference between a pixel and one of its neighboring pixels is greater than the value of TH_P_{DIFF}, this pixel is detected as a high gradient pixel. High gradient pixels are detected for Y, U, and V components separately, and the number of high gradient pixels $NumP_{HG}$ in a CU is set to the maximum number of the high gradient pixels detected by each component. Based on the value of $NumP_{HG}$, CUs are classified as high gradient CU (HGCUs) and non-high gradient CU (NHGCUs)

$$\text{CU} \in \begin{cases} \text{HGCU, if } NumP_{HG} > TH\_NumP_{HG} \\ \text{NHGCU, if } NumP_{HG} \leq TH\_NumP_{HG} \end{cases} \quad (2)$$

where the value of the threshold TH_NumP_{HG} is defined according to CU sizes of 2N×2N pixels

$$TH\_NumP_{HG} = \frac{2N\times2N}{64}. \quad (3)$$

Therefore, if the value of $NumP_{HG}$ in a CU is large than the value of TH_NumP_{HG}, the CU is defined as a HGCU. Otherwise, it is defined as a NHGCU.

The background color of a CU refer to the color with the highest occurrence frequency within the CU by considering all the three components. It is observed in Fig. 1 that the noiseless SCBs usually contain a large area with background color. On the contrary, the samples in NIBs are usually different because of noise. CUs are classified into background color CUs (BCCUs) and non-background color CUs (NBCCUs) by checking if the background color of a CU exists in its four sub-blocks with equal size

$$\text{CU} \in \begin{cases} \text{BCCU, if } BC \in \{S_1 \cap S_2 \cap S_3 \cap S_4\} \\ \text{NBCCU, if } BC \notin \{S_1 \cap S_2 \cap S_3 \cap S_4\} \end{cases} \quad (4)$$

where BC represents the background color of the current CU. It should be noted that $\{S_1 \cap S_2 \cap S_3 \cap S_4\}$ would contain BC only if all sub-CUs contain the background color of the current CU. Therefore, if the background color of a CU exists in all sub-blocks, the CU is defined as a BCCU. Otherwise, it is defined as a NBCCU.

To investigate the distributions of HGCUs *vs.* NHGCUs, and BCCUs *vs.* NBCCUs over different modes, "BasketballScreen" was encoded as an example, and the results with d = 2 and QP of 32 are shown in Fig. 5. It is observed that most CUs encoded by Intra mode are NHGCUs and NBCCUs, while most CUs encoded by IBC and PLT modes are HGCUs and BCCUs. Based on this observation, incoming CUs are finally classified as NIBs and SCBs

$$\text{CU} \in \begin{cases} \text{NIB, if CU} \in \{\text{NHGCUs} \cap \text{NBCCUs}\} \\ \text{SCB, if CU} \in \{\text{HGCUs} \cap \text{BCCUs}\} \\ \text{Uncertain CU, otherwise} \end{cases} . \quad (5)$$

Therefore, a CU is classified as a NIB if it is a NHGCU and NBCCU. On the contrary, a CU is classified as a SCB if it is a HGCU and a BCCU. Only Intra mode is checked for NIBs while only PLT and IBC modes are checked for SCBs. Otherwise, a CU is treated as an Uncertain CU and no early mode decision is made. Table I shows the average hit rate of the efficient mode decision technique for dynamic CUs over 14 test sequences in CTC [21]. The hit rate is calculated as the percentage of the CUs whose optimal modes from the original SCC encoder are not skipped using the proposed technique. It is observed that the hit rates for stationary CUs with d of 0 to 4 are 90.78%, 94.59%, 95.02%, and 94.63%, respectively, and it proves the high precision of the proposed efficient mode decision technique for dynamic CUs.

B. Efficient Mode Decision for Stationary CUs

Since screen content sequences contain many stationary CUs, the optimal mode correlation between the current CU and its collocated CU are utilized to make efficient mode decision.

To utilize the optimal mode correlation, an intuitive idea is to encode the current stationary CU with the same optimal mode as its collocated CU. However, we find that this approach induces high RD performance loss, and the results for the three reprehensive sequences are shown in Fig. 6. ΔTime and Bjøntegaard delta bitrate (BDBR) represent the encoding time increment and RD performance loss compared with the original encoder, respectively. It is observed that since "Kimono1" does not contain the stationary CUs, this approach has no impact on it. However, "Console" and "MissionControlClip3" suffer from 4.2% and 6.8% BDBR reduction. As reviewed in Section II, the selection of IBC mode and PLT mode is depended on the previous coded CUs.

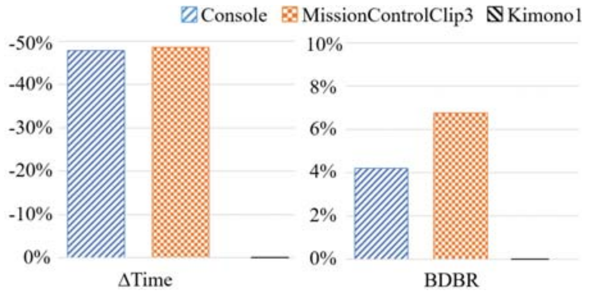

Fig. 6. The performance of encoding the stationary CU with the same optimal mode as the collocated CU.

TABLE II. AVERAGE HIT RATE OF THE PROPOSED EFFICIENT MODE DECISION TECHNIQUE FOR STATIONARY CUS

Hit Rate	d = 0 (%)	d = 1 (%)	d = 2 (%)	d = 3 (%)
Average	97.20	97.49	97.00	96.10

increment, although they have nearly 50% encoding time If the matching block for the current CU is disappeared in the current frame or the predicted palette table of the current CU is different from the collocated CU, the selection of IBC or PLT mode in the current CU can be different from the collocated CU. In this case, encoding the current stationary CU using the same optimal mode as the collocated CU leads to high RD performance loss. To eliminate this incorrect prediction, the efficient mode decision technique for stationary CUs is defined in two steps:

1) The optimal mode of the collocated CU is checked by the current CU.

2) If the RD cost of the current CU is larger than the collocated CU, the remaining modes are also checked. Otherwise, the optimal mode search of the current CU is terminated.

Since the inefficient IBC and PLT modes can be recognized by comparing the RD cost with the collocated CUs, the remaining modes are checked to reduce the RD performance loss. Table II shows the average hit rate of the efficient mode decision technique for stationary CUs over 14 test sequences in the common test condition (CTC) [21]. It is observed that the hit rates for stationary CUs with d of 0 to 4 are 97.20%, 97.49%, 97.00%, and 96.10%, respectively, and it proves the high precision of the proposed efficient mode decision technique for stationary CUs.

To achieve a higher encoding time reduction, the proposed efficient mode decision technique for dynamic CUs is also applied to stationary CUs if they need to check remaining modes after comparing the RD cost.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The proposed efficient mode decision algorithm has been implemented in the SCC reference software, SCM-8.3 [22]. To evaluate the performance of the proposed algorithm, the change in encoding time ΔTime and BDBR [23] with QPs of 22, 27, 32, and 37 have been compared with those of the original SCM-8.3 in percentage (%) under All Intra configuration. It should be noted that a negative value of BDBR or Δ Time denotes decrement in percentage as compared with SCM-8.3. The 14 sequences in CTC [21] are divided into four categories according to the video content, where TGM represents text and graphics with motion, M represents mixed content, A represents animation and CC represents camera-captured content. Since sequences in TGM and M are videos containing both NIBs and SCBs, while sequences in A and CC only contain NIBs, we will show the average results for sequences in TGM+M and A+CC, respectively. Besides, the performance of the proposed algorithm is compared with the existing fast SCC intra prediction algorithms, and they include fast CU partition algorithm proposed by Zhang *et al*. [13], fast CU partition and mode decision algorithms proposed by Duanmu *et al*. [16], Yang *et al*. [17], and Lei *et al*. [18]. To make fair comparisons, they were all re-implemented based on the same recent SCC reference software as our algorithm, SCM-8.3, for fair comparison.

The performance comparisons are shown in Table III. It is observed that the proposed algorithm outperforms the existing fast SCC intra prediction algorithms [13, 16-18]. On average, 40.05% encoding time reduction is provided with 1.40% increase in BDBR. For sequences in TGM and M, 42.43% encoding time is reduced with 1.63% increase in BDBR. For sequences in A and CC, 31.34% encoding time is reduced with 0.53% increase in BDBR. Zhang *et al*. [13] and Duanmu *et al*. [16] have similar increase in BDBR to the proposed algorithm. However, they only provides 33.19% and 26.89% encoding time reduction, respectively, which is much smaller than the proposed algorithm. Besides, it is observed that although Zhang *et al*. [13] works well for sequences with many stationary CUs, it reduces limited encoding time for sequences with many dynamic CUs, such as "FlyingGraphics", "Robot", "EBURainFruits", and "Kimono1", where only 4.6%, 12.04%, 16.48% and 0.46% encoding time is reduced, respectively. On the contrary, the proposed algorithm provides large encoding time for all sequences. For "FlyingGraphics", "Robot", "EBURainFruits", and "Kimono1", 21.97%, 28.50%, 28.43% and 37.08% encoding time reductions are provided. Yang *et al*. [17] and Lei *et al*. [18] induce much higher increase in BDBR than the proposed algorithm, which are 3.50% and 2.36%, respectively. Besides, their time savings are also smaller than the proposed algorithm, which are 35.36% and 33.20%, respectively. It is also observed that Yang *et al*. [17] induces very small increase in BDBR for sequences in A and CC. The reason is that it always checks Intra mode for all CUs with 2N×2N PUs to get features for classification. However, this process brings unnecessary computational overhead for SCBs because they will not be encoded by Intra mode. Comparatively, the proposed algorithm only utilizes two simple yet efficient features, high gradient pixels and background colors, to perform CU type classification. Therefore, the computational overhead of the proposed algorithm is much lower than Yang *et al*. [17], which helps to reduce encoding time.

V. CONCLUSION

In this paper, an efficient mode decision algorithm is proposed by content analysis. Incoming CUs are divided into stationary CUs and dynamic CUs by comparing their content with the collocated CUs. For a dynamic CU, the high gradient pixels and the background color are utilized to classify it as a NIB or a SCB. Then only Intra mode is checked for a NIB while only IBC and PLT modes are checked for a SCB to reduce the encoding time. For a stationary CU, the optimal mode of its collocated CU is checked. If its RD cost is larger than its collocated CU, remaining modes need to be checked. Otherwise, the search for optimal mode is terminated. The proposed algorithms has been implemented in the SCC

TABLE III. PERFORMANCE COMPARISONS WITH THE STATE-OF-THE-ART FAST INTRA PREDICTION ALGORITHMS

Sequences	**Zhang *et al.* [13]**		**Duanmu *et al.* [16]**		**Yang *et al.* [17]**		**Lei *et al.* [18]**		**Proposed Algorithm**	
	BDBR (%)	***∆Time (%)***	***BDBR (%)***	***∆Time (%)***	***BDBR (%)***	***∆Time (%)***	***BDBR (%)***	***∆Time (%)***	***BDBR (%)***	***∆Time (%)***
ChineseEditing (TGM)	0.65	-49.73	1.10	-17.47	4.30	-34.16	0.99	-18.96	0.53	-53.69
Console (TGM)	3.36	-39.35	1.87	-28.12	7.38	-42.83	2.87	-23.40	0.94	-48.35
Desktop (TGM)	1.95	-47.94	2.19	-26.24	6.27	-35.91	1.97	-23.85	1.58	-53.66
FlyingGraphics (TGM)	0.84	-4.60	0.98	-20.13	5.47	-31.19	1.72	-18.13	1.81	-21.97
Map (TGM)	0.85	-36.95	1.55	-19.16	2.84	-41.66	1.23	-20.05	2.65	-48.32
Programming (TGM)	1.16	-40.44	1.89	-22.16	4.71	-27.38	2.50	-22.92	1.09	-32.98
SlideShow (TGM)	1.39	-44.15	2.82	-52.47	3.69	-34.45	2.32	-55.58	2.55	-35.14
WebBrowsing (TGM)	2.05	-51.73	1.91	-28.17	5.00	-53.00	6.02	-26.75	1.57	-54.69
BasketballScreen (M)	1.06	-41.84	1.25	-22.43	3.00	-31.54	1.46	-24.83	2.10	-43.02
MissionControlClip2 (M)	1.29	-39.08	2.86	-33.9	2.51	-38.54	1.71	-25.49	1.91	-36.72
MissionControlClip3 (M)	1.05	-39.91	2.03	-24.61	2.90	-34.15	1.69	-33.81	1.25	-38.21
Robot (A)	0.92	-12.04	1.18	-29.36	0.59	-28.19	5.21	-46.91	1.38	-28.50
EBURainFruits (CC)	0.71	-16.48	0.88	-26.47	0.17	-25.89	1.76	-48.58	0.18	-28.43
Kimono1 (CC)	0.15	-0.46	1.23	-25.75	0.13	-36.18	1.52	-75.55	0.04	-37.08
Average (TGM+M)	1.42	-39.61	1.86	-26.81	4.37	-36.80	2.23	-26.71	1.63	-42.43
Average (A+CC)	0.59	-9.66	1.10	-27.19	0.30	-30.09	2.83	-57.01	0.53	-31.34
Average (ALL)	1.25	-33.19	1.70	-26.89	3.50	-35.36	2.36	-33.20	1.40	-40.05

reference software, SCM-8.3. Experiments results have shown that the proposed algorithm provides an average encoding time reduction of 40.05% with a negligible increase in BDBR of 1.40%, which outperforms four existing fast intra prediction algorithms of SCC.

REFERENCES

[1] Y. Lu, S. Li, and H. Shen, "Virtualized screen: A third element for cloud-mobile convergence," *IEEE Multimedia*, vol. 18, no. 2, pp. 4–11, February. 2011.

[2] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging hevc screen content coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 50–62, Jan. 2016.

[3] G. J. Sullivan, J. Ohm, W.-J. 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.

[4] X. Xu *et al.*, "Intra block copy in HEVC screen content coding extensions", *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp. 409–419, Dec. 2016.

[5] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Exploiting inter-layer correlations in scalable HEVC for the support of screen content videos," in *Proc. 19th Int. Conf. Digital Signal Process*, Hong Kong, China, Aug. 2016, pp.1–5.

[6] Z. Ma, W. Wang, M. Xu, and H. Yu, "Advanced screen content coding using color table and index map", *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4399–4412, Oct. 2014.

[7] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Hash based fast local search for intra block copy (IntraBC) mode in HEVC screen content coding," in *Proc. APSIPA ASC*, Hong Kong, Dec. 2015, pp. 396–400.

[8] S.-H. Tsang, W. Kuang, Y.-L. Chan and W.-C. Siu, "Fast HEVC screen content coding by skipping unnecessary checking of intra block copy mode based on CU activity and gradient," in *Proc. APSIPA ASC*, Jeju, Korea, Dec. 2016, pp.1–5.

[9] W. Kuang, S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast mode decision algorithm for HEVC screen content intra coding," in *Proc. of Int. Conf. on Image Process.*, Beijing, China, Sept. 2017, pp. 2473-2477.

[10] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Machine learning based fast intra encoding decision for HEVC screen content coding via decision trees," *IEEE Trans. Circuits Syst. Video Technol.*, early access, 2019.

[11] S.-H. Tsang, Y.-L. Chan, W. Kuang, and W.-C. Siu, "Mode skipping for HEVC screen content coding via random forest," early access, *IEEE Trans. Multimedia*, 2019.

[12] F. Duanmu, Z. Ma, and Y. Wang, "Fast CU partition decision using machine learning for screen content compression," in *Proc. IEEE Int. Conf. Image Process.*, Quebec, QC, Canada, Sep. 2015, pp. 4972–4976.

[13] H. Zhang, Q. Zhou, N.-N Shi, F. Yang, X. Feng, and Z. Ma, "Fast intra mode decision and block matching for HEVC screen content compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Shanghai, China, Mar. 2016, pp.1377–1381.

[14] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "Fast intraprediction for high-efficiency video coding screen content coding by content analysis and dynamic thresholding," *J. Electron. Imaging*, vol. 27, no. 5, pp. 053029-1-053029-18, Oct. 2018.

[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 Trans. Image Process.*, vol. 29, no. 1, pp. 170-185, January 2020.

[16] F. Duanmu, Z. Ma, and Y. Wang, "Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension," *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 6, no. 4, pp.517–531, Dec. 2016.

[17] H. Yang, L. Shen, and P. An, "An efficient intra coding algorithm based on statistical learning for screen content coding", in *Proc. IEEE Int. Conf. Image Process.*, Beijing, China, Sep. 2017, pp. 2468–2472.

[18] 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 Trans. Broadcast*., vol. 63, no.1, pp.48–58, Mar. 2017.

[19] S.-H. Tsang, W. Kuang, Y.-L. Chan and W.-C. Siu, "Reduced-complexity intra block copy (IntraBC) mode with early CU splitting and pruning for HEVC screen content coding," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp.269–283, Feb. 2019.

[20] W. Kuang, Y.-L. Chan, S.-H. Tsang, and W.-C. Siu, "DeepSCC: Deep learning based fast prediction network for screen content coding," *IEEE Trans. Circuits Syst. Video Technol.*, early access, 2019.

[21] H.-P. Yu, R. Cohen, K. Rapaka, and J. -Z Xu, "Common test conditions for screen content coding", 21th JCT-VC meeting, document JCTVC-U1015-r2, Warsaw, Poland, Jun. 2015.

[22] HM-16.12+SCM-8.3, HEVC test model version 16.12 screen content model version 8.3, [Online], available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.12+SCM-8.3/.

[23] G. Bjontegaard, "Calculation of average PSNR differences between rd-curves," document VCEG-M33, VCEG, Austin, Texas, USA, Mar. 2001.