Fast Palette Mode Decision Methods for Coding Game Videos With HEVC-SCC

Yu Liu, *Member, IEEE*, Cheng Fang, Jinglin Sun, and Xiangdong Huang, *Member, IEEE*

***Abstract*— The live broadcasting of game playing is becoming more and more popular. The game video is one of the typical video data which should be coded with high efficiency video coding-screen content coding (HEVC-SCC). In HEVC-SCC, the palette mode is an important technique which can effectively enhance the coding performance. However, this technique also demands high-coding computation. In this paper, we propose a fast palette mode pre-decision method based on the analysis of the color complexity of the video data. Considering only the coding computation of palette mode of HEVC-SCC, our algorithm saves about 74.24% computation in the experiments of game videos. Whereas, the BDBR only increases by about 0.36% and the BDPSNR decreases by about 0.03 dB on average.**

***Index Terms*— Game videos, palette mode, HEVC-SCC.**

I. Introduction

In ORDER to improve the coding efficiency of high definition videos, the ITU-T video coding experts group (VCEG) and ISO/IEC moving picture experts group (MPEG) formed a joint collaborative team (JCTVC) to develop the next generation video coding standard, which was named as “High Efficiency Video Coding” (HEVC) [1]. Though the HEVC standard can obtain good performance in many applications, the video coding tools provided by HEVC are primary suitable for natural video data captured by cameras. However, variable screen contents (SC) are coded and transmitted by different applications nowadays, such as PC desktop operation sharing, video conferencing, and media presentation on smartphones. In order to improve the coding efficiency of SC videos, the HEVC working group issued the special Call for Proposals (CfP) [2] for screen content coding (SCC) in January 2014, and some new coding tools were added into HEVC standards to form an extension: HEVC-SCC [3].

In the HEVC-SCC, a new coding tool named palette (PL) mode coding is very effective for SC videos, and some methods have been proposed to improve the performance of PL techniques. In [4], five techniques including Palette predictor propagation, Palette predictor stuffing, Palette sharing, Index map transposing, and Removing redundant copy are introduced. In [5], some needless modes and depths are skipped based on Sharp Edge Based Classification (SEBC) and Directional Edge Based Classification (DEBC). In [6], a fast intra prediction method is proposed based on content property analysis for HEVC-based SCC, in which the coding units (CUs) are classified into natural content CUs and screen content CUs. In [7], transition copy and prediction across coding unit boundary are proposed to improve color index map coding. In [8], a new palette copy mode named as COPY_INTER is proposed and applied to improve the performance of motion compensation.

Recently, many people share live-videos of their game playing online, and this kind of video content is becoming more and more popular. Limited by the power and computation resources of smartphones or laptops, high efficient PL coding methods are desired. However, existing fast HEVC coding algorithms are rarely designed for the game videos. In this paper, we suggest a pre-decision method for the PL mode in HEVC-SCC, and our algorithm can effectively enhance the coding efficiency.

The rest of the paper is organized as follows: Section II briefly reviews the theory and techniques of PL mode. Section III presents the details of our fast PL mode coding methods, and Section IV provides the experimental results of coding game videos and standard SC testing sequences with our proposals. Finally, Section IV concludes the paper.

II. Overview of the Palette Mode

Fig. 1 illustrates the encoding process of the palette-based coding method. In the PL mode coding, one mode flag is signalled at the CU level to indicate whether the palette mode is used. If a CU adopts the palette mode, it will select a set of major colors from the CU to form a palette table, and then a palette index map will be generated based on the palette table and CU data. The palette index map divides the pixels of the CU into major colors and escape colors, which are marked as blank blocks and patterned blocks respectively in Fig. 1. For the pixels whose colors exist in the palette table, only the indices in the palette are coded. Whereas, for the pixels whose colors do not exist in the palette table, their color values are considered as escape colors, and the quantized color values (if lossy coding is used) are directly encoded. Besides, two predictive coding modes: copy-left mode and copy-above

Manuscript received December 31, 2017; revised April 29, 2018 and July 25, 2018; accepted September 25, 2018. Date of publication October 4, 2018; date of current version October 2, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61771338 and Grant 61671012. This paper was recommended by Associate Editor J. Xu. *(Corresponding author: Xiangdong Huang.)*

Y. Liu, C. Fang, and J. Sun are with the School of Microelectronics, Tianjin University, Tianjin 300072, China (e-mail: liuyu@tju.edu.cn).

X. Huang is with the School of Electrical and Information School, Tianjin University, Tianjin 300072, China (e-mail: xdhuang@tju.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2018.2873700

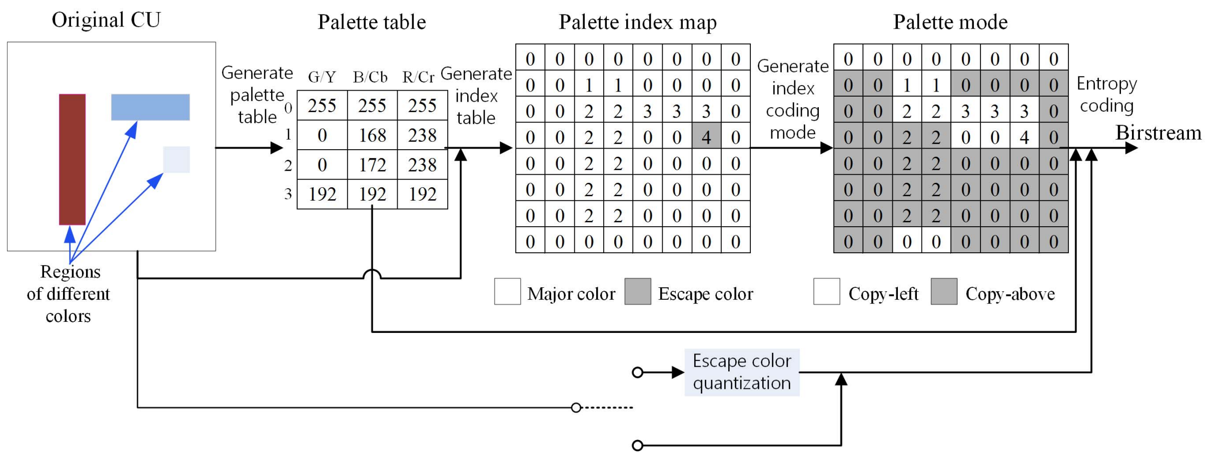


Fig. 1. The palette mode coding process of the HEVC-SCC.

mode are used to improve the coding efficiency of the palette index [9].

The JCTVC have provided techniques description and reference SC encoder in some reports, such as [10]. In the reference SC encoder, the color values of the original CU are clustered into k sets using modified k-means method, and each palette color is represented by a palette index (0, 1, 2, etc.). For each CU, the CU data is checked with the color of different palettes, and then the palette index map is formed for each CU (as shown in Fig. 1). In order to decrease the computation complexity of color checking, some methods such as predictive methods are accepted by JCTVC and applied in the reference encoder [11]–[13]. Though these methods can reduce some computations, the computation complexity of the PL mode is still very high.

As the PL techniques of HEVC-SCC can not obviously improve the coding efficiency of natural videos [14], directly applying the PL techniques to game videos coding may not achieve good performance because of the diversity of game contents. The game image can be roughly classified into three types: virtual nature images, animation images and smooth images. For virtual nature images, the PL techniques may not obtain good performance as these images contain many colors and details like natural images. Besides, some image regions only contain pure color blocks (smooth images), and these regions are also not suitable for the PL mode because some traditional coding methods can efficiently code these data [15].

In order to evaluate the coding performance of SC coding methods, JCTVC suggested the testing conditions of SCC in [16]. Besides, some algorithms have been suggested to improve the coding efficiency of SC coding. A fast-deciding CTU partition mode algorithm is proposed in [17] based on entropy and coding bits, and about 32% coding time can be saved. In [18], a fast intra coding method is proposed using smooth block judgments, and the intra coding complexity has been decreased by the fast method.

Most existing fast coding methods of SC video use the standard testing sequences or screen data of PCs. Whereas, most games use the animation to construct the contents, and these game videos always contain many regions with simple colors, which are very suitable for the PL techniques. Considering the game video coding on smartphones or laptops, methods which can apply the PL techniques according to the features of video contents should be developed, so that good balance between the coding complexity and quality can be achieved.

III. Proposed Methods

Generally, the PL techniques can enhance the coding efficiency mainly in regions which contain a few colors and a few simple textures. In other words, image regions which are too complex or too simple are not suitable for PL techniques (as discussed in [14] and [15]). In this paper, the regions suitable/unsuitable for PL technique are named as "PLM Regions" and "NPLM Regions" respectively. In order to derive a rough division of "PLM Regions", we suggest constructing a pre-decision process based on analyzing the color texture of the video data. The "PLM Regions" can be annotated by the pre-decision, and the computation can be decreased in regions which are not belong to the "PLM Regions".

A. DCT Based Color and Texture Analysis

In the color and texture analysis, we only use the luminance (Y) data to decrease the analyzing complexity. For the game videos which are always in 4:2:0 YUV format, the resolution of the Y data is higher than the chroma data (Cb, Cr). For other video frames such as 4:4:4 YUV and 4:4:4 RGB (can be converted into 4:4:4 YUV), the resolution of the Y data is same as the chroma data, thus analyzing Y data can also obtain desired results.

As shown in Fig. 2, the Y data of each video frame is divided into 8×8 blocks and transformed into the DCT domain, and DCT_{ij} represents the DCT coefficients (range between ± 2048) with i, j representing the row and column number respectively. As the traditional DCT can explore the textures of image data effectively, we adopt the traditional DCT using codes of [19]. Except for DCT_{00} which is the DC coefficient, other coefficients (AC coefficients) can reflect the variations of image data. Hence, our method uses AC coefficients to derive parameter K which could be used to evaluate the texture complexity of

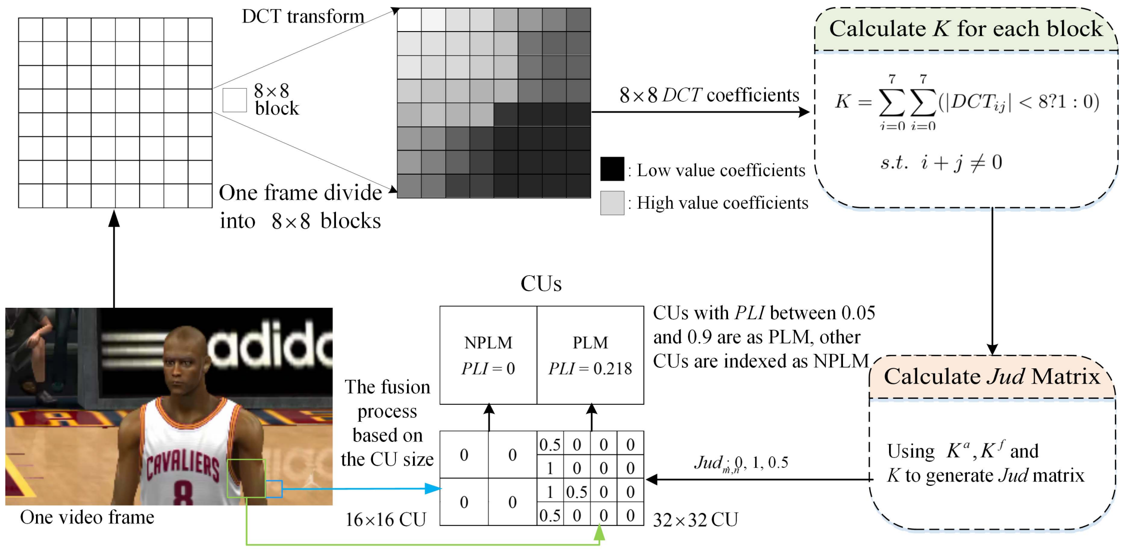

DCT transform
8×8 block
One frame divide into 8×8 blocks
8×8 DCT coefficients
: Low value coefficients
: High value coefficients
Calculate K for each block
$K = \sum_{i=0}^{7}\sum_{i=0}^{7}(|DCT_{ij}| < 8?1 : 0)$
$s.t.\ i+j \neq 0$
CUs
NPLM PLI = 0
PLM PLI = 0.218
CUs with PLI between 0.05 and 0.9 are as PLM, other CUs are indexed as NPLM
Calculate Jud Matrix
Using K^a, K^f and K to generate Jud matrix
$Jud_{m,n}$: 0, 1, 0.5
The fusion process based on the CU size
16×16 CU
32×32 CU
One video frame

Fig. 2. The flow chart of our algorithm.

image data:

$$K = \sum_{j=0}^{7}\sum_{i=0}^{7}(|DCT_{ij}| < 8?1 : 0) \quad s.t.\ i + j \neq 0 \tag{1}$$

The threshold 8 is selected as most AC coefficients with values lower than 8 will be quantized to zero in video coding, and these small coefficients always correspond to relatively pure color textures. Using Eq.(1), the texture features of data blocks could be reflected by K, and a high K corresponds to the block which contains smooth textures. Giving one video frame, the average K of one frame can be derived as:

$$K^f = \sum_{j=0}^{h-1}\sum_{i=0}^{v-1} K_{ij}/((v) \times (h)) \tag{2}$$

Here, h and v are the numbers of the horizontal and vertical 8×8 blocks in one frame respectively. We calculate the K^f of all the standard test sequences of HEVC-SCC in our experiments, and the average value of K^f of these sequences is denoted as K^a. Based on the statistic calculation of all test sequences, the K^a can be set as 55.

As the textures of the "PLM Regions" should not be too simple or too complex ([14], [15]), two thresholds: T_h and T_l are used in our method to judge the "PLM Regions". The high threshold T_h is used to judge smooth texture blocks which can be efficiently coded with the traditional Intra prediction modes such as the DC_mode of [15]. While, the low threshold T_l is used to detect the complex blocks. Generally, as most details might be maintained in the complex parts (corresponding to low K value) of the reconstructed images, T_l, which is used to identify the complex blocks, might be relatively stable. However, the simple texture parts (corresponding to high K value) of the reconstructed image are always smooth like a single color block, and the values of K may vary between a relatively bigger range. Thus, we adopt K^f and K^a together to select the T_h.

In order to obtain the optimal T_h and T_l, we carefully tune different values with all testing sequences, and use the grid-search strategy in a heuristic manner to obtain the best value. After our grid-search experiments, we find T_l can be set as 10% of the K^f (Eq.(3)). For T_h, the optimal value should be set more flexible because of the varying K, and we find the optimal value of T_h can be set with two factors (0.7 and 0.75) by K^f and K^a, as shown in Eq.(4).

It should be mentioned, the optimal parameters which can always obtain the best coding performance is hard to be selected, and the optimal value may vary for different SC videos. Thus, we only suggest a set of parameters which can achieve the averagely best performance for all the game videos and other standard testing sequences.

$$T_l = K^f \times 0.1 \tag{3}$$

$$\begin{cases} T_h = K^f \times 0.7 & if\ K^f > K^a \\ T_h = K^f \times 0.75 & if\ K^f \leq K^a \end{cases} \tag{4}$$

Further, K of each 8×8 block is compared with T_h and T_l to generate the normalized complexity parameter for each 8×8 block (Jud_{mn}):

$$Jud_{mn} = \begin{cases} 0, & K_{mn} > T_h \\ 0.5, & T_l \geq K_{mn} > T_h \\ 1, & K_{mn} \leq T_l \end{cases} \tag{5}$$

Here, the m and n represent the row and column number corresponding to the location of the 8×8 blocks in one frame, and the color complexity of one video frame can be represented by a matrix constructed by Jud_{mn}. For each 8×8 block, $Jud_{mn} = 1$ means the color complexity of the block is very high.

B. Composition of the PL Index for Different CU Sizes

In HEVC coding, several sizes of CUs are supported, thus the Jud_{mn} matrix which is based on the 8×8 unit can not be directly adopted to judge whether the PL mode should be

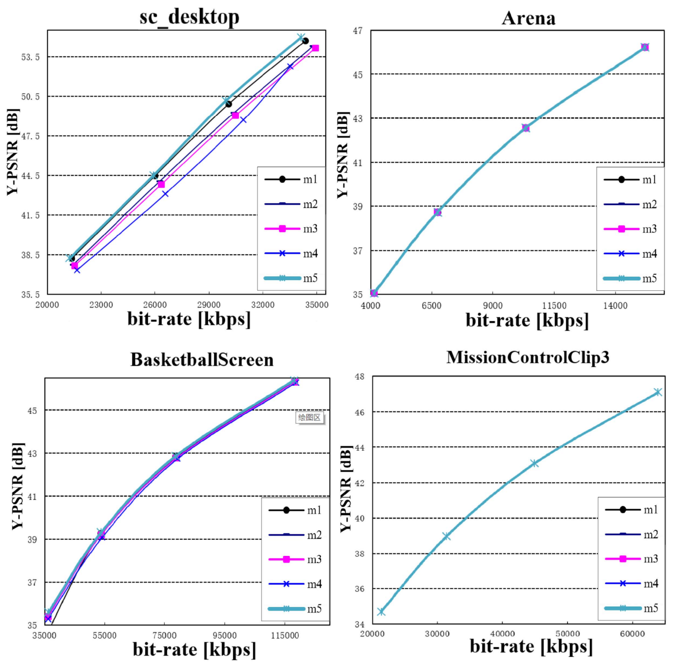


Fig. 3. Coding Results of Different PLI ranges: m1 = 0.15-0.8, m2 = 0.25-0.7, m3 = 0.3-0.65, m4 = 0.4-0.55, m5 = 0.05-0.9.

used for one CU. In our method, we compose the values of Jud_{mn} based on the size of CU to generate the PL indexes (PLI) for CUs with size of $H \times V$:

$$\begin{cases} PLI = \frac{1}{N}\sum_{i=0}^{V/8-1}\sum_{j=0}^{H/8-1} a_{mn} \times Jud_{mn} \\ a_{ij} = |DCT_{00}^{mn}|/(\frac{1}{N}\sum_{i=0}^{V/8-1}\sum_{j=0}^{H/8-1}|DCT_{00}^{mn}|) \end{cases} \tag{6}$$

Here, $N = H \times V/8 \times 8$ is the number of 8×8 blocks in one CU, a_{mn} is the weight of each Jud_{mn}, and the a_{mn} is derived by the DCT_{00} which could reflect the average color of each 8×8 block. Using Eq.(6), a PLI representing the color complexity of each CU can be derived, and the PLI values are used to enable/disable the PL mode in the HEVC-SCC process. Take 16×16 CU for example, there are four 8×8 blocks in the CU, which means there are four corresponding Jud_{mn} values. The PLI is the weighted (a_{mn}) values of the four Jud_{mn}, and the value of PLI is between 0 and 1.

As discussed in [14] and [15], blocks which are too complex or too simple are not suitable for PL techniques. In our proposal, we adopt several ranges for PLI to find the optimal range which can effectively classify the complexity of blocks. Fig. 3 shows the results of some sequences with different ranges. For some sequences such as Arena and MissionControlClip3, the coding performances between different ranges (m1 to m5) are trivial. But in some other cases (such as sc_desktop and BasketballScreen), the different ranges obviously impact the coding qualities. Based on the grid-search experiments using all test sequences, the range of PLI is set as m5 $= \{0.05, 0.9\}$. In other words, when $PLI < 0.05$ or $PLI > 0.9$, the PL techniques are not used in our proposed HEVC-SCC encoder. Though the thresholds may

TABLE I
PARAMETERS OF GAME-VIDEO AND STANDARD TESTING SEQUENCES

Sequence name	Resolution	FPS	YUV	Type
XXL	1920×1080	25	4:2:0	Smartphone
Asphalt8CC	1280×720	25	4:2:0	Smartphone
Arena	1280×720	25	4:2:0	Smartphone
NBA2K17	1280×720	25	4:2:0	PC
ChinaSpeed	1024×768	30	4:2:0	PC
Honkai3	640×480	25	4:2:0	Smartphone
BasketballScreen	2560×1440	60	4:4:4	PC
EBURainFruits	1920×1080	50	4:4:4	PC
Kimono1	1920×1080	24	4:4:4	PC
MissionControlClip2	2560×1440	60	4:4:4	PC
MissionControlClip3	1920×1080	60	4:4:4	PC
sc_desktop	1920×1080	60	4:4:4	PC
sc_flyingGraphics	1920×1080	60	4:4:4	PC
sc_Programming	1280×720	60	4:4:4	PC
sc_Robot	1280×720	30	4:4:4	PC
sc_Slideshow	1280×720	20	4:4:4	PC
sc_Web_Browsing	1280×720	30	4:4:4	PC
sc_Console	1920×1080	60	4:4:4	PC
sc_Map	1280×720	60	4:4:4	PC

not guarantee the best performance for some special videos, these values have achieved the averagely best performance in all the game videos and other standard testing sequences.

In summary, we propose a pre-decision process to enale/disable the PL techniques of HEVC-SCC as shown in Fig. 2. The texture complexity of one 8×8 block is estimated by parameter K (Eq.(1)). Then, two threshold (T_h and T_l) are carefully derived by analyzing K of many testing sequences, and T_h and T_l are further used to generate normalized complex parameter Jud for each 8×8 block. At last, PLI for each CU is composed by $Juds$ according to the size of CU. Using PLI as a index to enale/disable the PL techniques on the CU level, the HEVC-SCC coding complexity can be saved.

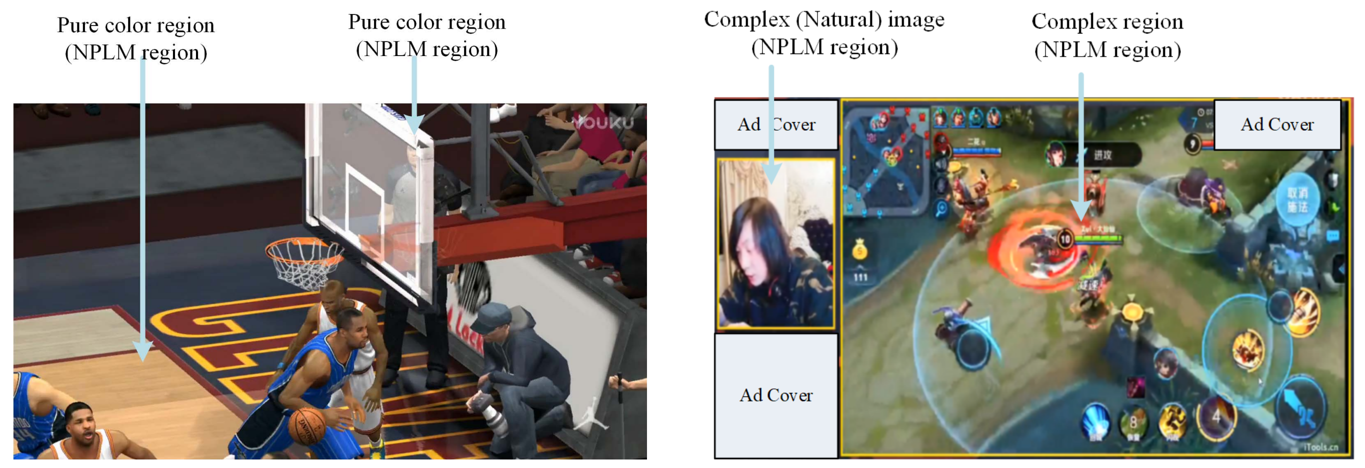
IV. Experimental Results and Discussions

The experiments of our proposed algorithm are implemented on the HM-16.6+SCM-5.0 (REF encoder), and the hardware platform is Intel Core i7 CPU-2.67GHz, 6G RAM. The coding configuration is set as all Intra coding. We also build a Game-video sequence set as listed in Table I (in [20]), and some contents of these game videos are shown in Fig. 4 (Some areas showing advertisements in Fig. 4(b) are intentionally covered by "Ad Cover" blocks).

Table II shows the experimental results of our proposed algorithm on game videos. In Table II, the REF encoder with PL enabled is denoted as REF_PL, whereas the REF coder with PL disabled is denoted as REF_NPL, and the TS denotes the average encoding time saving:

$$TS = \frac{|T - T_{ori}|}{T_{ori}} \times 100\% \tag{7}$$

Here, T_{ori} represents the total encoding time of the REF_PL, and T means the total encoding time of our method or the REF_NPL. The T_{ori} or T is accounted by all coding process (the pre-processing of our proposal is included). For the DCT



(a) (b)

Fig. 4. Two examples of the game test sequences: (a) NBA2K17, and (b) Arena.

TABLE II
PERFORMANCE OF THE PROPOSED ALGORITHM COMPARED WITH THE REF CODER IN GAME VIDEOS

Sequences	Proposed / REF_PL			REF_NPL / REF_PL		
	BDPSNR/dB	BDBR/%	TS/%	BDPSNR/dB	BDBR/%	TS/%
XXL	0	-0.04	22.48	0.01	-0.16	23.41
Asphalt8CC	-0.04	0.55	20.34	-0.08	1.04	26.14
Arena	-0.01	0.13	18.77	-0.14	1.64	25.03
NBA2K17	-0.01	0.11	18.21	-0.02	0.29	19.18
ChinaSpeed	-0.11	1.28	14.81	-0.68	8.12	28.59
Honkai3	-0.01	0.14	17.49	-0.02	0.27	28.58
Average	**-0.03**	**0.36**	**18.68**	**-0.15**	**1.87**	**25.16**

TABLE III
PERFORMANCE OF THE PROPOSED ALGORITHM COMPARED WITH THE REF CODER IN STANDARD VIDEOS

Sequences	Proposed / REF_PL			REF_NPL / REF_PL		
	BDPSNR/dB	BDBR/%	TS/%	BDPSNR/dB	BDBR/%	TS/%
BasketballScreen	-0.07	0.73	7.38	-1.03	11.77	12.28
EBURainFruits	0.00	0.00	8.42	0.00	-0.07	10.7
MissionCtrlClip2	-0.03	0.29	7.69	-1.27	12.65	12.21
MissionCtrlClip3	-0.05	0.45	6.58	-1.22	11.23	10.21
Kimono1	0.00	0.03	11.21	0.00	-0.05	13.02
sc_Robot	0.00	-0.09	15.93	0.01	-0.27	16.52
sc_Programming	-0.07	0.55	6.12	-1.90	17.80	8.64
sc_SlideShow	-0.12	1.68	12.49	-1.13	15.92	16.78
sc_flyingGraphics	-0.24	1.64	0.13	-2.07	15.54	2.81
sc_desktop	-0.50	1.40	2.24	-9.13	38.59	2.30
sc_Web_Browsing	-0.13	0.88	3.56	-3.21	29.59	5.59
sc_Console	-1.93	7.20	1.30	-0.80	39.51	5.46
sc_Map	-0.26	3.15	6.96	-0.85	11.70	10.00
Average	**-0.26**	**1.38**	**6.93**	**-1.68**	**15.68**	**9.73**

processing, which consumes the main computation of the pre-processing in our proposal, it only accounts for less than 1% of T (rough analysis using the profiling tools of VS 2013). As shown in Table II, averagely, the proposed algorithm obtains about 18.68% time saving (compared using the total coding time) when applied to game video sequences, whereas the BDBR increases about 0.36% and the BDPSNR decreases about 0.03 dB. Concerning the processing of the PL techniques in REF_PL, our proposal can save about 74.24% PL coding complexity.

It can be noticed that for some games like the "NBA2K17", which is shown in Fig. 4(a), the coding performance difference between REF_NPL and REF_PL is not obvious as this sequence contains many pure color regions. According to our method, most PLI of these regions are smaller than 0.05, so our proposed algorithm can effectively decrease the coding complexity. For videos which contain natural images and complex color regions (such as 4(b)), our method can also disable the palette mode as the most PLI of these regions are bigger than 0.9.

Table III also presents the performance of our algorithm on standard sequences recommended by the JCTVC [16]. The results show that the proposed algorithm achieves about 6.93% time saving in the total coding process, whereas the BDBR increases by about 1.38% and the BDPSNR decreases by about 0.26 dB. Compared with the REF PL, our proposals also can save 71% coding complexity of PL techniques.

For the accuracy of the PL coding mode pre-decision, we can only take the mode decision of REF_PL as the ground truth. In the coding process of REF_PL, coding of current blocks always use the surrounding reconstructed data as reference, and the mode decision is related with many factors such as the modes of surrounding blocks. Hence, giving one block with different reference data, the mode decision may be different, and the differences will drift in the following coding blocks. Thus, the ground truth of PL mode in REF_PL can not be mapped accurately to the modes of our proposals, and we can only use the data of Table II and Table III to evaluate our proposals.

Besides, Table IV shows the comparison between our proposal and some prominent fast coding methods of HEVC (Ref. [6], [17], [18]). Though the coding methods of [17] and [18] are not designed for the PL techniques, the TS and coding performance of these fast algorithms can be used as references of single coding tool optimization, such as the fast intra coding methods of [18]. We believe our methods can be integrated with other algorithms to obtain better performance. Besides, though some of the references use different HM coder

TABLE IV
PERFORMANCE OF THE PROPOSED ALGORITHM IN [6], [17], AND [18] METHODS

Sequences	Proposed algorithm			Ref. [6]			Ref. [17]			Ref. [18]		
	BDPSNR/dB	BDBR/%	TS/%	BDPSNR/dB	BDBR/%	TS/%	BDPSNR/dB	BDBR/%	TS/%	BDPSNR/dB	BDBR/%	TS/%
BasketballScreen	-0.07	0.73	7.38	-0.15	1.61	37.21	-0.16	1.71	22.79	-0.02	2.64	6.33
EBURainFruits	0.00	0.00	8.42	-0.09	1.72	48.05	-0.01	0.15	13.92	0.00	0.01	0.39
MissionControlClip2	-0.03	0.29	7.69	-0.11	1.30	40.72	-0.06	0.74	27.42	-0.03	0.35	4.99
MissionControlClip3	-0.05	0.45	6.58	-0.09	0.82	31.51	-0.02	0.18	16.78	-0.04	0.33	2.60
Kimono1	0.00	0.03	11.21	-0.04	1.80	78.91	-0.01	0.28	22.88	0.00	0.02	1.00
sc_Robot	0.00	-0.09	15.93	-0.23	5.98	53.30	-0.01	0.34	14.37	-0.02	0.21	5.00
sc_Programming	-0.07	0.55	6.12	-0.13	1.16	35.87	-0.30	2.68	25.80	-0.03	0.28	5.72
sc_SlideShow	-0.12	1.68	12.49	-0.22	3.00	62.30	-0.07	0.95	47.02	-0.03	0.38	26.30
sc_flyingGraphics	-0.24	1.64	0.13	-0.15	1.03	31.12	-0.39	2.26	17.85	-0.25	1.67	4.20
sc_desktop	-0.50	1.40	2.24	-0.37	1.10	36.10	-0.86	2.58	20.45	-0.74	2.24	7.90
sc_Web_Browsing	-0.13	0.88	3.56	-0.40	2.63	39.00	-0.19	1.19	23.84	-0.34	2.15	12.96
Average	**-0.11**	**0.69**	**7.44**	**-0.18**	**2.01**	**44.91**	**-0.19**	**1.19**	**23.01**	**-0.14**	**0.93**	**7.04**

TABLE V
RESULTS OF IMAGE QUALITY SUBJECTIVE ASSESSMENT

Sequence name	Source	REF_PL	REF_NPL	Proposed
XXL	4.06	3.93	3.87	4.00
Asphalt8CC	4.33	3.73	3.67	3.07
Arena	4.13	3.20	3.20	3.07
NBA2K17	4.33	3.60	3.47	2.80
Honkai3	4.73	3.67	3.33	3.40
BasketballScreen	3.73	3.67	3.40	3.47
MissionControlClip2	4.87	3.53	3.73	4.00
MissionControlClip3	4.13	4.07	4.00	4.00
sc_desktop	5.00	5.00	5.00	5.00
sc_flyingGraphics	4.60	4.67	4.67	4.67
sc_Programming	3.73	3.87	3.26	4.67
sc_Robot	4.60	3.27	3.80	3.80
sc_Slideshow	4.93	4.93	4.93	4.93
sc_Web_Browsing	4.20	3.67	3.20	3.40
sc_Console	4.80	4.75	4.19	4.71
sc_Map	4.93	4.88	4.50	4.50
Average	**4.44**	**4.02**	**3.88**	**3.97**

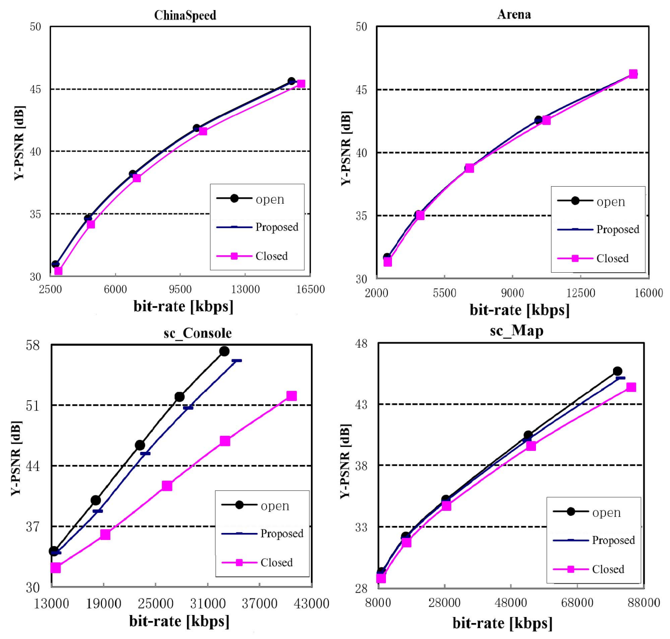


Fig. 5. R-D Curves of some testing sequences.

(HM-15.0+SCM-2.0), the SCC coding parts of these coders are same as our REF encoder (HM-16.6+SCM-5.0).

It can be noticed that [6] obtains the highest TS performance as it adopts several fast algorithms apart from the PL techniques. Whereas, our method is only designed for fast PL pre-decision, and the TS of our method only can reflect the computation decreasing in the PL mode coding. Generally speaking, though our proposed algorithm can not achieve more TS than some existing methods, our method outperforms all the references in terms of the BDPSNR. Theoretically, the TS of our method should be lower than the TS of REF_NPL, as our method is specially designed for the PL techniques.

Though our proposal has not improved the coding efficiency and quality significantly in the standard SC testing sequences, our methods obtained good performance in game videos. As other existing methods (such as [6], [17], and [18]) have not provided open-access coding software, it is very hard for us to complete fair comparing experiments between these methods and our approach, and we believe our methods can be integrated with other algorithms to obtain better performance in SC coding of game videos.

Considering the game videos as typical data of HEVC-SCC, REF_NPL obtains about 25.16% time saving when it is compared with REF_PL, as shown in Table II. In other words, the PL mode selecting process accounts for about 25.16% coding complexity of the whole coding processes. Using our proposal, the PL mode selection can be early decided, and our methods can save about 18.68% coding time when it is compared with REF_PL (Table II). Thus, our proposal saves about 74.24% PL coding complexity, and the BDPSNR only decreases about 0.03 dB. Whereas, the REF_NPL results in 0.15 dB (Table II) BDPSNR degradation. In summary, the proposed method is very suitable for coding game videos with HEVC-SCC, especially in the case of broadcasting game playing with smartphones which are limited by power and computation resources.

As the subjective quality is also very important for screen content video coding, we also complete experiment for the subjective quality assessments. Images of the source sequences, reconstructed image of our proposal, REF_PL, and REF_NPL are used as testing samples. Each image sample is displayed for 5 seconds on a 4K displayer, and a 3-seconds black image is used as the interval for each sample. 15 volunteers are invited to assess the sample images with MOS scores from 1-5. As shown by the MOS scores of Table V, the reconstructed images of our proposals also achieve high quality in the subjective assessment. Besides, some bit-streams and images used in the subjective assessment are also shared in [21].

V. Conclusion

In this paper, we propose a pre-decision method for HEVC-SCC based on analysis of color complexity of video data. First, 8×8 DCT transform is applied to image blocks, and the color complexity is evaluated by the DCT coefficients. Then, judge matrices are derived to make the pre-decision for the palette mode of HEVC-SCC. In order to evaluate the performance of our method, test sequences of typical game videos are built as Open-Access. According to the experimental results on these game videos, the proposed method can obtain about 74.24% computation saving, whereas the BDBR (+0.36%) and BDPSNR (-0.03 dB) are nearly not changed.

Acknowledgment

The authors would like to thank the anonymous reviewers for their comments and suggestions.

References

[1] *ISO IEC MPEG-H Part 2: High Efficiency Video Coding*, document H.265, 2013.

[2] *Joint Call for Proposals for Coding of Screen Content*, document I. JTC1/SC29/WG11 and M. ITU-T Q6/SG16, San Jose, CA, USA, Jan. 2014.

[3] R. Joshi and J. Xu, *Screen Content Coding Draft Text 1*, document JCTVC-R1005, Joint Collaborative Team Video Coding, Sapporo, Japan, Jul. 2014.

[4] Y. C. Sun *et al.*, "Palette mode—A new coding tool in screen content coding extensions of HEVC," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 2409–2413.

[5] Y. Kawakami, G. Chen, and T. Ikenaga, "Content based mode and depth skipping with sharp and directional edges for intra prediction in screen content coding," in *Proc. IEEE 12th Int. Colloq. Signal Process. Appl.*, Mar. 2016, pp. 46–49.

[6] 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.

[7] Y.-C. Sun *et al.*, "Improved palette index map coding on HEVC SCC," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 4210–4214.

[8] W. Zhu *et al.*, "Inter-palette coding in screen content coding," *IEEE Trans. Broadcast.*, vol. 63, no. 4, pp. 673–679, Dec. 2017.

[9] P. Onno, X. Xiu, Y. W. Huang, and R. Joshi, *Suggested Combined Software and Text for Run-Based Palette Mode*, document JCTVC-R0348, Joint Collaborative Team Video Coding, Sapporo, Japan, Jul. 2014.

[10] R. Joshi, J. Xu, R. Cohen, S. Liu, Z. Ma, and Y. Ye, *Screen Content Coding Test Model 2 Encoder Description (SCM 2)*, document JCTVC-R1014, Joint Collaborative Team Video Coding, Sapporo, Japan, Jul. 2014.

[11] Y. C. Sun *et al.*, *SCCE3 Test a.2: Palette Coding and Palette Predictor Updates*, document JCTVC-R0119, 18th Meeting, Joint Collaborative Team Video Coding, Sapporo, Japan, Jun. 2014.

[12] P. Lai, S. Liu, T. D. Chuang, Y. W. Huang, and S. Lei, *Non-RCE4: Major Color Table (Palette) Sharing*, document JCTVC-P0153, 16th Meeting, Joint Collaborative Team Video Coding, San Jose, CA, USA, Jan. 2014.

[13] C. Gisquet, G. Laroche, and P. Onno, *SCCE3: Test a.3—Palette Stuffing*, document JCTVC-R0348, 18th Meeting, Joint Collaborative Team Video Coding, Sapporo, Japan, Jun. 2014.

[14] 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.

[15] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[16] H. Yu, R. Cohen, K. Rapaka, and J. Xu, *Common Test Conditions for Screen Content Coding*, document JCTVC-R1015, Joint Collaborative Team Video Coding, Sapporo, Japan, Jul. 2014.

[17] M. Zhang, Y. Guo, and H. Bai, "Fast intra partition algorithm for HEVC screen content coding," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 390–393.

[18] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast and efficient intra coding techniques for smooth regions in screen content coding based on boundary prediction samples," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 1409–1413.

[19] *OpenDVDx*. Accessed: 2017. [Online]. Available: https://sourceforge.net/p/opendvdx/code-0/114/tree/trunk/DVDx/src/libmpeg/fdctref.cpp

[20] *Game-Video Sequence*. Accessed: 2017. [Online]. Available: https://pan.baidu.com/s/1HujXPGWBlwLYvVMq9pHINg

[21] *Experimental Data*. Accessed: 2017. [Online]. Available: https://pan.baidu.com/s/1B iYVc87obMMp7bKECd4vkA

Yu Liu (S'02–M'06) received the B.Eng. degree in electronic engineering, the M.Eng. degree in information and communication engineering, and the Ph.D. degree in signal and information processing from Tianjin University, Tianjin, China, in 1998, 2002, and 2005, respectively. From 1998 to 2000, he was an Electronic Engineer with Nantian Electronics Information Corporation, Shenzhen, China. From 2011 to 2012, he was a Visiting Fellow with the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. He is currently a Professor with the School of Microelectronics, Tianjin University. His research interests include signal/video processing, medical signal processing, machine learning, compressed sensing, and indoor positioning system.

Cheng Fang received the B.Eng. degree in electronic information engineering from Tianjin University, Tianjin, China, in 2015, where he is currently pursuing the M.S. degree in information and communication engineering. His research interests focus on algorithms of video coding.

Jinglin Sun received the B.Eng. degree in communication engineering from Dalian Ocean University, Dalian, China, in 2016. She is currently pursuing the Ph.D. degree in circuits and systems with Tianjin University, Tianjin, China. Her current research interests include machine learning and intelligent video analysis.

Xiangdong Huang is currently an Associate Professor with the School of Electrical and Information, Tianjin University. His research interests include signal processing, machine learning, and compressed sensing.