The following publication S. Tsang, Y. Chan, W. Kuang and W. Siu, "Reduced-Complexity Intra Block Copy (IntraBC) Mode With Early CU Splitting and Pruning for HEVC Screen Content Coding," in IEEE Transactions on Multimedia, vol. 21, no. 2, pp. 269-283, Feb. 2019, doi: 10.1109/TMM.2018.2856078 is available at https://doi.org/10.1109/TMM.2018.2856078.

Reduced-Complexity Intra Block Copy (IntraBC) Mode with Early CU Splitting and Pruning for HEVC Screen Content Coding

Sik-Ho Tsang, *Member, IEEE*, Yui-Lam Chan, *Member, IEEE*,
Wei Kuang, *Student Member, IEEE*, and Wan-Chi Siu, *Life Fellow, IEEE*

***Abstract*— A Screen Content Coding (SCC) extension to High Efficiency Video Coding (HEVC) has been developed to incorporate many new coding tools in order to achieve better coding efficiency for videos mixed with camera-captured content and graphics/text/animation. For instance, Intra Block Copy (IntraBC) mode helps to encode repeating patterns within the same frame while Palette mode aims at encoding screen content with few major colors. However, IntraBC mode brings along high computational complexity due to the exhaustive block matching within the same frame though there are already some constraints and fast approaches applied to IntraBC mode to reduce its complexity. Thus, we propose a fast intra coding scheme to reduce the complexity of using IntraBC mode in SCC. Screen content always contains no sensor noise resulting in the characteristics with pixel exactness along both horizontal and vertical directions. These characteristics pave the way for mode skipping and early Coding Unit (CU) splitting. Besides, early CU pruning and early termination are proposed based on the Rate Distortion (RD) cost to further reduce the encoder complexity. Moreover, we also propose to reduce the complexity of IntraBC mode by checking the hash value of each block candidate and the current block during block matching. With our proposed scheme, the encoding time is reduced compared with the SCC while the coding efficiency can still be maintained with minor increase in Bjontegaard Delta Bitrate (BDBR).**

***Index Terms*—hash search, HEVC, intra block copy, screen content coding, video coding.**

I. Introduction

With the rapid development of network bandwidth and widespread usage of thin-client devices, computer screen sharing has become much popular today. Applications include remote desktops, video conferencing with documents or PowerPoint slides sharing, and even cloud computing which provides cloud services by screen sharing technology [1]. Screen Content Coding (SCC) is highly demanded for limited network bandwidth and has emerged as one of the hot research topics in the aspect of video coding. Thus, there was a Call for Proposal (CfP) of SCC as the extension of High Efficiency Video Coding (HEVC) [2] by the Joint Collaborative Team on Video Coding (JCT-VC) in January 2014 [3]. As a result, SCC has been introduced as the extension of HEVC [4-6] for coding screen content.

Screen content videos are typically computer-generated content such as text, computer graphics and graphic user interface, or the compound of camera-captured content and computer-generated content. The camera-captured content can be efficiently encoded by the conventional HEVC. However, the computer-generated content has discontinuous-tone characteristics, such as complex structure with sharp edges [7-11], which are different from those of camera-captured content. As usual, computer-generated content is also composed of only a limited number of colors and sometimes contains high contrast between colors. Thereby, the conventional HEVC intra prediction [12-13], which uses the neighboring boundary pixels for prediction, cannot handle well these kinds of content. Another characteristic of screen content is that repeating patterns, with a variety of shapes and sizes such as texts, icons, control buttons, menus, slider bars, window frames, and graphs, frequently occur within the same frame. There is also a noiseless characteristic that all pixels within a block can be exactly equal along the horizontal or vertical directions. In some cases, all pixels within the block are even exactly the same. Therefore, numerous research works based on these characteristics have been suggested for SCC.

A. Sharp Edges

Screen content with a complex structure, a limited number of colors and sharp edges are well encoded by Palette mode [14-17], which separates the screen content into two parts of information: color information and structural information. The color information is represented by a few base colors while the structural information is represented by a palette index map where each palette index specifies the color of each pixel. Base colors are predicted using a palette table. If the color cannot be predicted by the palette table, it is encoded as an escape color. For palette index coding, run-length based techniques, copy-left mode and copy-above mode, are used.

B. Noiseless Areas

Unlike camera-captured content, screen content does not contain any camera noise because it is not captured by a camera

Manuscript received October 2016. The work described in this paper was a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Grant No. PolyU 152016/14E).

S.-H. Tsang, Y.-L. Chan, W. Kuang, and W.-C. Siu, are with the Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: sik-ho.tsang@polyu.edu.hk, enylchan@polyu.edu.hk, wei.kuang@connect.polyu.hk, enwcsiu@polyu.edu.hk).

but generated by computer software. Thus, there are plentiful noiseless areas and only one single color appears within the area. Zhang *et. al* [18] suggested to have a new Independent Uniform Prediction (IUP) mode in which a set of colors is stored in the slice header. For the whole block with only one single color, an index is coded for the IUP mode to choose the color from the set of colors. The decoder then fills that color into the block based on the index. But it is not HEVC compliant. The work in [19] proposed to decide whether the block needs to be further split based on estimated entropy and coding rate. If the entropy or the coding rate of the block is smaller than pre-defined thresholds, no further split is decided for the block in order to speed up the encoding process. On the other hand, our previous work [20] proposed to handle the noiseless area by Simple Intra Prediction (SIP) which is a simplified version of the conventional intra prediction [12]. By using SIP, Rough Mode Decision (RMD) [13] and Rate Distortion Optimization (RDO), which include checking of numerous intra prediction candidates, are skipped for reducing the complexity of Intra mode. However, the above approaches [18-20] do not consider the case of all pixels within a block being exactly equal along the horizontal or vertical directions.

C. Repeating Patterns

For repeating patterns in screen content, intra Motion Estimation (ME) and Motion Compensation (MC) were developed in many research works [7-8,21-39]. In [7-8,21-25], arbitrary string matching in 1D and 2D forms with variable lengths and shapes was suggested. In [26-28], intra line copy was proposed to use fixed-length line matching during intra ME and MC for better tradeoff between coding efficiency and complexity. In addition, Intra Block Copy (IntraBC) mode was proposed in [29-38] to perform rectangular block matching within the same frame. It is proven that intra ME and MC can significantly improve the coding efficiency for SCC [29-30]. Recently, IntraBC mode has been adopted in the SCC extension [6]. It is considered as an additional mode besides Intra mode and Palette mode in intra coding. Exhaustive block matching is done for IntraBC wherein the Sum of Absolute Difference (SAD) is estimated between every block candidate and the current block. This is a very time consuming task compared with Intra mode and Palette mode. Thereby, Budagavi *et. al* [29-30] proposed to apply some constraints and a fast encoding technique to reduce the complexity of IntraBC. Full-frame hash search was also proposed in [31-34] where block matching is only done for those block candidates which have the same hash value as the current block being encoded. Zhu *et al.* [32-33] suggested to estimate the hash value based on the DC value of a CU as well as the number of color transitions along the row and column, while Li *et al.* [34] proposed to estimate the hash value based on the Cyclic Redundancy Check (CRC) value of an CU to find the identical repeating patterns in hash search. Other research works [35-38] also proposed four non-compliant modes to further improve the coding efficiency: flipping, symmetric, masking, and rotational modes respectively.

D. Previous Work and Our Contributions

However, Palette mode and IntraBC mode induce huge complexity in SCC, which is demonstrated in Table I. The YUV 4:4:4 scene content sequences shown in this table were encoded using the reference software HEVC Test Model Version 16.7 Screen Content Model Version 6.0, hereafter called SCM-6.0 for the sake of simplicity. Table I tabulates the Bjontegaard Delta Bitrate (BDBR) and encoding time against HEVC in percentage for various sequences, where HEVC+SCC (Palette Only) denotes the conventional HEVC SCC extension with the use of Palette mode only, and HEVC+SCC denotes the conventional HEVC SCC extension with the use of both Palette and IntraBC modes. For HEVC+SCC (Palette Only), there is 28.92% of BDBR reduction with the average encoding time increased by 33.57% compared with HEVC. For HEVC+SCC, 43.58% of BDBR reduction is obtained with the average encoding time increased substantially by 94.03% compared with HEVC. It can be seen that Palette mode provides a better RD improvement with less complexity surge compared with IntraBC mode. The high complexity of IntraBC mode is mainly due to the exhaustive block matching.

To reduce the SCC encoder complexity, the work in [40] suggests to exploit the CU depth and mode from the collocated CUs of the previous frame for stationary CUs. In addition, an adaptive step-size search was proposed for speeding up IntraBC block matching. This scheme mainly handles the stationary CUs. A machine learning scheme with three classifiers using decision trees was proposed in [41]. The first classifier divides CUs into camera-captured content CUs (CCCUs) for skipping IntraBC and Palette modes, and screen content CUs (SCCU) for skipping the conventional Intra mode. The second classifier then splits CCCUs into partitioned CUs and non-partitioned CUs for fast CU partitioning. Non-partitioned CUs are further classified as directional CUs or non-directional CUs. Based on the classification result, different types of CUs then go through reduced set of intra prediction directions. Similarly, the algorithm in [42] proposed to classify CUs into CCCUs and SCCUs and then utilize content property analysis, bits per pixel information as well as neighboring and collocated CUs' depth information to perform mode elimination and fast CU partitioning.

In this paper, we focus on the complexity reduction in SCC

TABLE I
BDBR (%) AND ENCODING TIME (%) OF HEVC+SCC (PALETTE ONLY) AND HEVC+SCC AGAINST HEVC

Sequences	HEVC+SCC (Palette Only)		HEVC+SCC	
	BDBR (%)	Time (%)	BDBR (%)	Time (%)
Basketball_Screen	-29.48	+42.02	-50.55	+100.92
MissionControlClip2	-21.34	+47.17	-46.63	+106.14
MissionControlClip3	-37.02	+37.78	-65.33	+96.43
ChineseEditing	-49.22	+34.92	-58.95	+117.39
sc_flyingGraphics	-34.94	+30.81	-61.21	+93.60
sc_desktop	-60.70	+23.06	-83.16	+74.63
sc_console	-56.95	+18.19	-68.76	+56.96
sc_web_browsing	-54.14	+26.82	-79.19	+71.74
sc_map	-14.83	+46.97	-20.78	+160.95
sc_programming	-32.27	+39.07	-52.11	+83.77
sc_SlideShow	-14.36	+26.57	-22.17	+58.83
sc_robot	+0.29	+47.01	-1.33	+120.06
EBURainFruits	+0.05	+25.07	-0.06	+97.71
Kimono1	+0.03	+24.47	+0.05	+77.34
Average	-28.92	+33.57	-43.58	+94.03

by expediting IntraBC block matching, early splitting and pruning the SCC tools based on the screen content characteristics or information. The differences between our contributions and the recent schemes [40-42] can be summarized as: 1) Our work is different from [40] which heavily relies on stationary region detection in the current frame and reuses the CU depth information from the stationary regions in the previous frame. This makes [40] only efficient in stationary CUs while our proposed algorithm can work for both stationary CUs and non-stationary CUs; 2) Unlike [41] and [42], which firstly classify CUs into CCCUs and SCCUs, our work only focus on SCCUs. The techniques for SCCUs proposed in this paper could be easily incorporated into the framework in [41] and [42] for handling CCCUs; and 3) Different from [40] and [41], our proposed algorithm is self-contained and does not need information from other frames. When it is feasible to use the modes and CU depth information from the previous frame, the proposed algorithm can also work with the CU depth prediction to further speed up the SCC encoder.

The remainder of this paper is organized as follows. We firstly start by introducing the intra coding of SCC extension. Then, we proceed to describe our proposed reduced-complexity scheme based on the image characteristics of the block. Section III discusses how the number of IntraBC mode checking can be reduced, and how the decision on the block splitting and pruning can be made, as well as the number of search points within the IntraBC block matching can be lessened. Finally, experimental results for our proposed scheme and its integrations with other recent schemes are shown in Section IV followed by conclusions in Section V.

II. HEVC SCC Extension Intra Coding

In this section, we describe the related part of intra coding in the SCC extension that has been developed in the reference software SCM-6.0 [43-44].

In general, each video frame is divided into Coding Tree Units (CTUs) with the size of $L{\times}L$ where L can be chosen as 64, 32, 16 or 8. CTUs in a frame are encoded in raster scan order. A recursive quad-tree coding structure is applied to each CTU [45]. Each CTU can be split into four smaller Coding Units (CUs) of equal size, namely sub-CUs or child sub-CUs. Each sub-CU can be further split into four smaller CUs of equal size in which this CU splitting is processed recursively (The supported CU size is from 64×64, 32×32, 16×16, down to 8×8). The encoder then chooses the optimum CU size by comparing the Rate Distortion (RD) cost obtained by the current CU and the sum of RD costs obtained by the four sub-CUs. In SCC intra coding, the mode that obtains the least RD cost among the Intra, Palette and IntraBC modes, is selected as the best mode. The encoder decides to split the CU if the sum of the costs of the four sub-CUs, $J_l(t)$, is smaller than the cost of the current CU, J_{2l}:

$$\sum_{t=0}^{3} J_l(t) < J_{2l} \tag{1}$$

where l and t are the sub-CU size and the index for each sub-CU, respectively. To compute the RD cost, the distortion between the original CU and the reconstructed CU, and the coding rate are required to be estimated.

For each $2N{\times}2N$ CU, where $2N$ is the size of that particular CU (i.e. N can be 32, 16, 8 or 4), it can be further partitioned into different sizes of Prediction Unit (PU) of $2N{\times}N$, $N{\times}2N$ and $N{\times}N$. The encoder attempts to search for the best PU size within that particular CU in which the least RD cost is obtained. As a result, the encoding complexity of a CTU is large because there is additional Palette mode and IntraBC mode for each CU candidate in SCC intra coding.

A. *Intra Mode and Palette Mode*

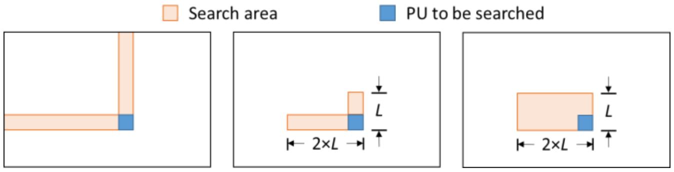
For Intra mode in SCM-6.0, boundary pixels are used for predicting a CU with 33 directional predictions plus planar and DC predictions [12]. RMD [13] is performed to select a subset of intra prediction candidates first. Then the optimal one is chosen by RDO where the full RD cost for every candidate in the subset is estimated. In SCM-6.0, only $2N{\times}2N$ and $N{\times}N$ are supported for Intra mode. For Palette mode [14-17], base colors and the corresponding index map are estimated and encoded using the palette table and run-length coding, respectively, as mentioned in the previous section. It is noted that only $2N{\times}2N$ are used for Palette mode. A detailed technical overview of Palette mode is provided in [17].

B. *Intra Block Copy (IntraBC) Mode*

For IntraBC mode, block matching using intra ME and MC is carried out within the same frame. If IntraBC is used by one particular CU, each PU within the CU is encoded with a Block Vector (BV) as well as the residual signal of that CU [39]. To perform block matching for one search point, SAD between the CU candidate and the current CU, $SAD(x,y)$, as well as the cost of BV, $BVCost(x,y)$, are estimated to be the IntraBC cost, $IBCCost(x,y)$, as a total cost in the following:

$$IBCCost(x,y) = SAD(x,y) + BVCost(x,y) \tag{2}$$

where (x,y) is the position of the CU candidate. From (2), the SAD and BV costs are estimated once for every search point, and the one with the minimum $IBCCost(x,y)$ is selected as the optimal candidate. For BV cost estimation, it occupies very small amount of time compared with SAD calculation as it is just simply estimated by a look-up table plus few operations. The complexity of block matching mainly comes from the SAD calculation. It should be noted that the computational complexity of IntraBC mode is much higher than those of Intra and Palette modes due to the exhaustive block matching. Several constraints for balancing the coding efficiency and computational complexity were thus suggested in [29], which has been implemented in SCM-6.0. First, BVs are limited to be integer-pel accurate instead of fractional-pel accurate. With this restriction, interpolation process can be skipped. Second, pre-defined area search is recommended such that full-frame search is not required. Last, IntraBC mode is only applied for small CU sizes of 16×16 and 8×8, since repeating patterns usually appear for small CU sizes rather than large CU sizes. To find the matched block, there are pre-defined area search and hash search.



(a) (b) (c)

Fig. 1 Illustrations of (a) full vertical and horizontal searches, (b) vertical and horizontal pre-defined area searches, and (c) 2D pre-defined area search.

1) Pre-defined Area Search for IntraBC

Different CU and PU sizes have different searching strategies in IntraBC. Fig. 1 illustrates the search areas for various searching strategies including the full vertical and horizontal searches, vertical and horizontal pre-defined area searches, and 2D pre-defined area search. For 16×16 CUs, only 2*N*×2*N* PUs with full vertical and horizontal searches are performed, as depicted in Fig. 1(a). It is due to the fact that large CU size tends to have fewer repeating patterns found within the same frame.

For 8×8 CUs, if it is a *N*×2*N* PU, only full vertical and full horizontal searches are performed. If it is 2*N*×2*N* or 2*N*×*N*, vertical, horizontal and 2D pre-defined area searches within the left and current CTUs are carried out, as shown in Fig. 1(b) and Fig. 1(c). In SCM-6.0, the 2D search for 2*N*×2*N* is skipped if the CU activity, Act, is smaller than a pre-defined threshold TH_1 [30]. Act is estimated as follows:

$$\begin{gathered} Act = min(Act_H, Act_V) \\ Act_H = \sum_{j=0}^{7}\sum_{i=1}^{7}|p(i,j) - p(i-1,j)| \\ Act_V = \sum_{j=1}^{7}\sum_{i=0}^{7}|p(i,j) - p(i,j-1)| \end{gathered} \tag{3}$$

where $min(a,b)$ is the operation to choose the minimum value of a and b, and $p(i,j)$ is the luminance component of the pixel at the position (i,j) in the CU.

Similarly, the 2D search for 2*N*×*N* is not performed if the CU activity is smaller than TH_1 and the best RD cost so far is smaller than the pre-defined threshold TH_2. TH_1 and TH_2 in SCM6.0 are set as:

$$\begin{gathered} TH_1 = 168 \times (1 \ll BitDepth - 8) \\ TH_2 = max(66 \times \lambda, 800) \end{gathered} \tag{4}$$

where ≪ represents the left shift operation, $BitDepth$ is the bit depth of the pixel, $max(a,b)$ is the operation to choose the maximum value of a and b, and λ is the Lagrange multiplier depending on the Quantization Parameter (QP). Prior to doing the above searches, the CU candidates with positions pointed by a set of BVs are checked by (2) for CU sizes smaller or equal to 32×32 where the set of BVs includes two last coded BVs as well as neighboring coded BVs. In addition, the candidates are also checked by (2) using the BV predictors of skip and merge modes for all CU sizes before the IntraBC block matching. These two steps are not time consuming because no exhaustive block matching is required.

2) Full-frame Hash Search for IntraBC

For hash search, block matching is only done for those CU candidates which have the same hash value as the current CU to be encoded. By taking the tradeoff between the computational complexity and coding performance into consideration, hash search is only performed on 8×8 CUs with 2*N*×2*N* PU. The hash value is a 16-bit value calculated based on the DC of four sub-partitions of the CU, DC_k with $k=0,1,2,3$, and gradient of the whole CU, *Grad*, as follows:

$$\begin{gathered} DC_k = \frac{1}{4\times 4}\sum_{j=0}^{3}\sum_{i=0}^{3} p(i + (k\%2)\times 4, j + Floor(k/2)\times 4) \\ Grad = \sum_{j=1}^{7}\sum_{i=1}^{7}\frac{|p(i,j) - p(i-1,j)| + |p(i,j) - p(i,j-1)|}{2} \end{gathered} \tag{5}$$

where % is the modulo operator and *Floor(a)* is the floor function to give the largest integer less than or equal to *a*. The hash value is then formed by concatenating DC_k with $k=0,1,2,3$, and *Grad* as follows:

$$\begin{aligned} Hash = {} & (MSB_3(DC_0) \ll 13) + (MSB_3(DC_1) \ll 10) \\ & + (MSB_3(DC_2) \ll 7) + (MSB_3(DC_3) \ll 4) \\ & + MSB_4(Grad) \end{aligned} \tag{6}$$

where $MSB_m(X)$ means that the *m* Most Significant Bits (MSB) of *X* are extracted. By (5) and (6), the hash value is formed for each 8×8 CU candidate with 2*N*×2*N* PU. Thus, after the pre-defined area search, full-frame hash search is performed where block matching is only done for those search points with the same hash value as the current CU. After that, hash values of all the newly search points within the reconstructed area are estimated and can be re-used for the coming CUs. This means that CU candidates within the same frame but with different hash values from the current CU are filtered out. In other words, full-frame search can be carried out with less computational complexity by adopting the hash search.

C. Mode Decision Process for Intra Coding

To encode a CU, firstly if the CU size is 32×32 or smaller, as aforementioned, the CU candidates of IntraBC mode are checked by the set of BVs including two last coded BVs as well as neighboring coded BVs. If there is distortion, 2*N*×2*N* and *N*×*N* Intra modes are checked. Next, the candidates are checked using the BV predictors of skip and merge modes. If skip mode is chosen as the best mode so far, the encoding process of a CU is finished. Otherwise, for the case of 64×64 CU, the encoding process is finished. For 32×32 CU, only palette mode is checked. For 16×16 CU, 2*N*×2*N* IntraBC mode and Palette mode are checked and the encoding process of a CU is finished. For 8×8 CU, 2*N*×2*N* IntraBC mode is checked first. If the best RD cost so far is larger than or equal to a pre-defined threshold TH_3 estimated in SCM-6.0 as (7), *N*×2*N* IntraBC mode is checked. If the best RD cost so far is still larger than or equal to TH_3, 2*N*×*N* IntraBC mode is checked. Finally, Palette mode is checked and the encoding process of a CU is finished.

$$TH_3 = max(60 \times \lambda, 56) \tag{7}$$

Fig. 2 shows the simplified intra coding in SCC. After finished encoding a CU, the encoder splits the CU into four smaller sub-CUs for encoding, or encoding the next CU.

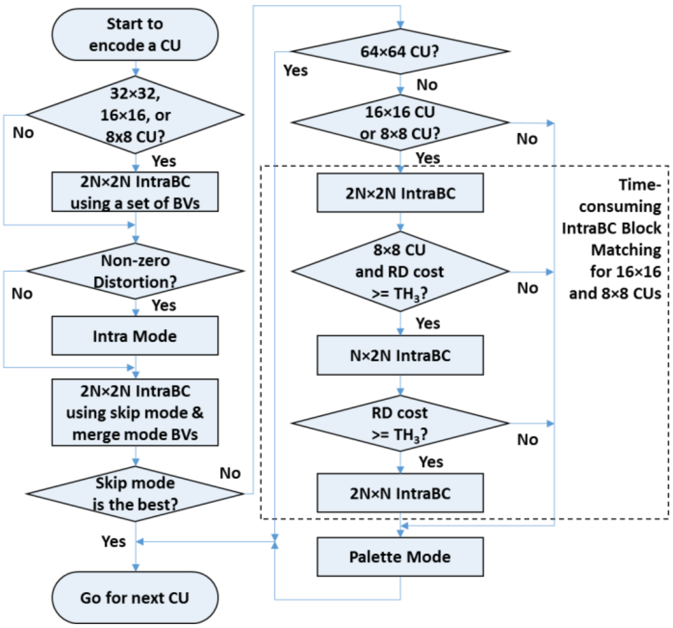


Fig. 2 Mode decision process in HEVC SCC extension intra coding.

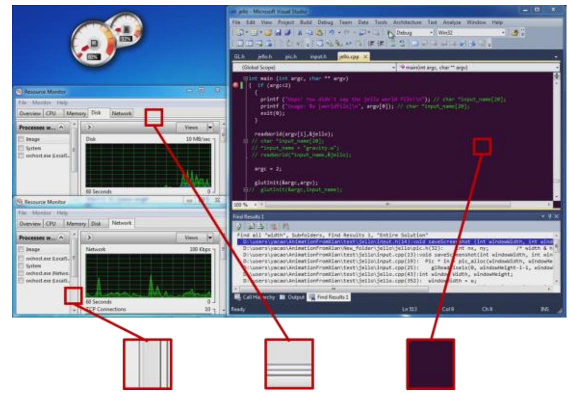


Fig. 3 Illustration of *sc_programming* screen content sequence where areas highlighted by red color can be perfectly reconstructed by either DC, horizontal or vertical intra predictions in conventional Intra mode.

III. Proposed Fast HEVC SCC Intra Coding Scheme

In this section, our fast intra coding scheme in SCC based on the local image characteristics and the RD cost of the current CU is proposed. Our previous work in [46] has initially designed an efficient hash based local search for IntraBC. In this paper, we further extend our work in [46] by skipping the checking of IntraBC mode and early CU splitting based on the pixel exactness of the CU. Early CU pruning and skipping IntraBC checking based on the RD cost is also suggested in order to reduce the complexity induced by IntraBC. Finally, based on the work in [46], detailed analysis on hash based pre-defined area search is investigated.

A. Skipping IntraBC and Palette Modes with Early CU Pruning Based on Pixel Exactness

Screen content (e.g., graphics, graphical user interface, text, etc.) has noiseless characteristics without camera noise. For some areas of screen content, it can be perfectly predicted by the conventional Intra mode without any distortion. If so, the IntraBC mode can be skipped to reduce the coding complexity. Fig. 3 illustrates an example of the screen content sequence *sc_programming* with areas highlighted by red color. In these areas, pixels in the same row or column are exactly the same, or all pixels within the CU are exactly the same. This kind of CU actually can be perfectly predicted by either DC, horizontal or vertical intra prediction in the conventional Intra mode. The CU is then defined to have the property of horizontal pixel exactness if the following condition is satisfied:

$$p(i,j) = p(0,j) \quad \forall p(i,j) \in \{P \mid i \geq 1\} \qquad (8)$$

where P is the set of pixels within the CU. Similarly, CU is defined to have the property of vertical pixel exactness if the following condition is satisfied:

$$p(i,j) = p(i,0) \quad \forall p(i,j) \in \{P \mid j \geq 1\} \qquad (9)$$

TABLE II
Number of CUs (%) that has property of horizontal, vertical or 2D pixel exactness

Sequences	Type	CU Size			
		64×64	32×32	16×16	8×8
Basketball_Screen	M	15.34	25.87	38.83	51.12
MissionControlClip2	M	9.08	15.42	25.29	35.83
MissionControlClip3	M	5.39	11.24	20.88	33.96
ChineseEditing	TGM	3.52	10.11	22.50	38.70
sc_flyingGraphics	TGM	1.83	14.61	35.09	57.22
sc_desktop	TGM	4.37	17.43	36.86	58.11
sc_console	TGM	1.66	15.94	44.30	69.42
sc_web_browsing	TGM	12.22	31.10	47.63	63.96
sc_map	TGM	9.71	13.40	16.43	23.82
sc_programming	TGM	10.09	20.27	32.56	45.59
sc_SlideShow	TGM	45.19	57.72	62.50	66.49
sc_robot	A	0.00	0.00	0.00	0.14
EBURainFruits	CC	0.00	0.00	0.00	0.00
Kimono1	CC	0.00	0.00	0.00	0.00
Average (M+TGM)		10.76	21.19	34.81	49.47
Average (A+CC)		0.00	0.00	0.00	0.05

In addition, the CU is defined to have the property of 2D pixel exactness if both (8) and (9) are satisfied. Table II tabulates the statistics about the percentage of CUs which has the property of horizontal, vertical or 2D pixel exactness using the YUV 4:4:4 sequences recommended in the Common Test Conditions (CTC) for SCC [47]. It can be seen that the percentages of horizontal, vertical or 2D pixel exactness of various CU sizes for Animation (A) and Camera Captured (CC) sequences are equal to or near to 0% due to camera noise and complex texture. In contrast, there are significant number of CUs in different sizes that have horizontal, vertical or 2D pixel exactness for screen content sequences with Text and Graphics with Motion (TGM), and Mixed (M) content. The percentages of 16×16 and 8×8 CU sizes that fulfill the property of horizontal, vertical or 2D pixel exactness increases to 34.81% and 49.47%, respectively. A CU fulfilling one of the above properties can be well predicted by both the conventional Intra mode and IntraBC mode with zero residual error. With zero residual error, the luminance and chrominance prediction directions are required to encode the CU in the conventional Intra mode while only the skip mode index for the skip mode, or the BV difference for the non-skip mode is required in IntraBC mode. The IntraBC mode requires fewer bits than the

TABLE III
BDBR (%) AND ENCODING TIME (%) WHEN INTRABC MODE IS SKIPPED IF CU HAS THE PROPERTY OF HORIZONTAL, VERTICAL OR 2D PIXEL EXACTNESS AGAINST HEVC+SCC

Sequences	Type	BDBR (%)	Encoding Time (%)
Basketball_Screen	M	+0.28	-7.22
MissionControlClip2	M	+0.08	-4.27
MissionControlClip3	M	+0.06	-3.43
ChineseEditing	TGM	+0.00	-5.40
sc_flyingGraphics	TGM	+0.28	-3.80
sc_desktop	TGM	+0.35	-8.46
sc_console	TGM	+0.38	-7.50
sc_web_browsing	TGM	-0.12	-4.41
sc_map	TGM	+0.11	-5.61
sc_programming	TGM	+0.21	-6.50
sc_SlideShow	TGM	+0.32	-31.95
sc_robot	A	+0.02	-0.48
EBURainFruits	CC	+0.00	+0.26
Kimono1	CC	+0.00	+0.02
Average (M+TGM)		+0.18	-8.05
Average (A+CC)		+0.01	-0.07
Average (All)		+0.14	-6.34

conventional Intra mode. An analysis shown in Table III has been conducted to show the average BDBR increase for various sequences when the conventional Intra mode replaces the IntraBC mode in the case of the CU with the property of horizontal, vertical or 2D pixel exactness. In Table III, only negligible BDBR increase of 0.14% is incurred with time reduction of 6.34%. Based on this analysis, we can conclude that the bit consumptions using the conventional Intra and IntraBC modes are very close, but the computational demanding IntraBC mode can be avoided. Hence, we propose that if the CU has the property of either horizontal, vertical or 2D pixel exactness, the checking of IntraBC and Palette modes can be skipped, and CU splitting can be terminated.

Besides, as the horizontal, vertical or DC modes can perfectly predict the CUs with the properties of pixel exactness, exhaustive calculations of RDO for all intra prediction candidates in the conventional RMD become unnecessary, since zero residual error has already been obtained by either horizontal, vertical or DC mode. Therefore, most candidates in RMD can also be skipped to reduce the complexity of the conventional Intra mode. In the proposed algorithm, for the case of 2D pixel exactness, only the first Most Probable Modes (MPM) prediction is checked. For the cases of horizontal and vertical pixel exactness, MPMs are not included, only horizontal and vertical predictions are checked, respectively, in the conventional Intra mode.

B. Early CU Splitting Based on Partial 2D Pixel Exactness

Normally, as in (1), the RD cost of the current CU (parent CU) and the RD costs of the four sub-CUs (child sub-CUs) are estimated recursively to determine the best combination for each CTU. This recursive quad-tree encoding approach is extremely time consuming since it involves determining the best mode of every CU with various sizes, especially in the checking of the most time consuming IntraBC mode for every CU that involves a large amount of SAD estimation, as mentioned in (2). Therefore, it is desired to have early CU splitting by taking the CU size of the parent CU and the 2D pixel exactness of its child sub-CUs into consideration. When a child sub-CU has the property of 2D pixel exactness, its residual error becomes zero, resulting in a very small RD cost, $J_l(t)$ in (1). The more the child sub-CU in the current CU fulfills the property of 2D pixel exactness, the higher the probability of the current CU to be split, since $\sum_{t=0}^{3} J_l(t)$ in (1) gets smaller. Therefore, we propose to have early CU splitting when there are X_{CU_SIZE} sub-CUs with the property of 2D pixel exactness, and the current CU is defined to have partial 2D pixel exactness. X_{CU_SIZE} can be 1, 2, or 3 where *CU_SIZE* is 64×64, 32×32 or 16×16.

TABLE IV
NUMBER OF CUS (%) THAT HAS PROPERTY OF PARTIAL 2D PIXEL EXACTNESS

X_{CU_SIZE} \ *CU_SIZE*	Average (M+TGM)			Average (A+CC)		
	64×64	32×32	16×16	64×64	32×32	16×16
1	21.77	32.92	50.78	0.00	0.00	0.14
2	18.72	27.32	42.71	0.00	0.00	0.03
3	9.73	15.59	24.54	0.00	0.00	0.00

In Table IV, the percentage of CUs with the property of partial 2D pixel exactness for different values of X_{CU_SIZE} is shown. Again, for (A+CC) sequences, the percentages of partial 2D pixel exactness are nearly 0% no matter when X_{CU_SIZE}=1, 2, or 3 for all CU sizes. For (M+TGM) sequences, those percentages of X_{CU_SIZE}=3 for all CU sizes are comparatively higher, from 9.73% to 24.54%. When X_{CU_SIZE} decreases, those percentages are even higher, from 21.77% to 50.78%. It is worth performing early CU splitting for CU based on the property partial 2D pixel exactness such that the corresponding RD cost estimation and the mode decision process of the current CU can be skipped. In this case, the number of checking IntraBC mode can also be reduced resulting in lessening the encoder complexity.

Instead of having a fixed value of X_{CU_SIZE} for all CU sizes, different X_{CU_SIZE} for each CU size is adopted in the proposed early CU splitting decision. This is because the decision also depends on the size of the current CU, since SCC mode decision for different CU sizes undergoes different processes, as described in Section II.C. In Fig. 2, IntraBC block matching is only applied to 16×16 and 8×8 CUs while only a simple pre-defined set of BVs and skip/merge modes of IntraBC is used in 32×32 and 64×64 CUs. Skipping the IntraBC block matching for 32×32/64×64 CUs is due to the reason that the repeating patterns rarely appear in these large CUs. It means that J_{2l} in (1) gets larger. Consequently, for 32×32/64×64 CUs, a smaller value of X_{CU_SIZE} might be sufficient to trigger the decision in (1) for early CU skipping. In comparison with 32×32/64×64 CUs, the chance of a parent CU with the size of 16×16 to find a perfect match or good match in the pre-defined area search via the efficient IntraBC block matching process is larger, J_{2l} in (1) gets smaller. A larger value of X_{CU_SIZE} might be needed for 16×16 CU. Detailed analysis for selecting values of X_{CU_SIZE} in the early CU skipping decision is discussed in Section IV.A.

C. Early CU Pruning and Skipping IntraBC Checking Based on RD Cost

In addition, early CU pruning is also proposed. If the RD cost estimated for one particular CU is smaller than a pre-defined threshold, further CU splitting is not processed. Consequently, the complexity induced by (1) and (2) can be reduced. To facilitate this pruning process, we propose to swap the coding order of Palette mode and IntraBC block matching process, as shown in Fig. 2. This is because the time consumed by checking the Palette mode is shorter than by checking the IntraBC mode. If the best RD cost so far before the IntraBC mode block matching is smaller than a pre-defined threshold TH_4, the exhaustive IntraBC mode block matching can be avoided. TH_4 is a tradeoff parameter to control the balance between the computational complexity and the coding efficiency, and it is defined as:

$$TH_4 = max(n \times \lambda, 1.5 \times n) \quad (10)$$

where n is the control factor to adjust TH_4. When the value of TH_4 is small, the probability of early CU pruning gets smaller, and the computational complexity is less reduced. In the case of a large TH_4, the probability of early CU pruning becomes larger, and the computational complexity is much reduced at the expense of coding efficiency. Therefore, the maximum operation is to ensure TH_4 is not too large to preserve the coding efficiency, and the concept is very similar to the case of TH_2 in (4) and TH_3 in (7) implemented in SCM-6.0. We empirically tested the values of TH_4 by varying the values of control factor, n, as described in Section IV.A.

D. Hash Based Pre-Defined Area Search for IntraBC mode

The complexity of pre-defined area search and hash search is analyzed by coding the testing sequences using SCM-6.0 [43-44] with All Intra (AI) configuration and QP={22, 27, 32, 37}, as recommended in the CTC for SCC [47]. We collected the statistics of the number of search points on the testing sequences with the same settings. Table V tabulates the number of search points for pre-defined area search and hash search in percentage for all sequences. It can be observed that the pre-defined area search occupies about 87.95% in average in IntraBC mode, while the hash search only contributes about12.05% for the complexity. Hence, it is highly motivated to reduce the complexity for the pre-defined area search.

To reduce the computational complexity for the pre-defined area search, we propose to reduce the number of search points to speed up the pre-defined area search by calculating the hash value in PU. Fig. 4 illustrates the idea of our proposed hash based fast pre-defined area search using the *sc_programming* sequence. In this figure, the black rectangle is the current PU being encoded while the red and green regions are the corresponding search areas if full vertical and horizontal searches are used. However, the possible relevant search points, indicated by the green region, are very few while there are a lot of irrelevant search points indicated by the red region. A hash value of a PU is computed before the pre-defined area search is carried out. IntraBC block matching is only carried out for those PU candidates with the same hash value as the current PU. In other words, if the hash values of the PU candidate and the current PU are the same, it is more likely that this PU candidate has similar color and structure of the current PU, and it belongs to the green region of Fig. 4. On the contrary, if their hash values are not the same, those PU candidates most likely exhibits different colors and structure of the current PU, and it is in the red region of Fig. 4. Thus, the SAD calculation of this PU candidate can be skipped to speed up the pre-defined area search. As a result, irrelevant search area is not searched and the encoding time can be reduced. By doing so, a large amount of search points can be skipped if the hash values of PU candidates and the current PU are checked first.

Yet, we cannot directly apply the hash value estimated by (5) and (6) into the pre-defined area search for checking whether the hash values of the PU candidates and the current PU are the same. One of the reasons is that the conventional full-frame hash search is used for 8×8 CU with 2N×2N PU only. On the other hand, the pre-defined area search can be used for 16×16 CU with 2N×2N PU as well as 8×8 CU with 2N×2N, 2N×N and N×2N PUs. The second reason is due to different coverage of the search areas used by the full-frame hash search and pre-defined area search. For the full-frame hash search, all search points within the reconstructed area in the frame are estimated, which supports long-distance repeated patterns [31]. This large search area requires a 16-bit hash value to filter out more search points. On the other hand, the search points in

TABLE V
NUMBER OF SEARCH POINTS (%) OF PRE-DEFINED AREA SEARCH AND HASH SEARCH

Sequences	Type	Pre-defined Area Search	Hash Search
Basketball_Screen	M	84.26	15.74
MissionControlClip2	M	86.64	13.36
MissionControlClip3	M	90.40	9.60
ChineseEditing	TGM	86.14	13.86
sc_flyingGraphics	TGM	78.14	21.86
sc_desktop	TGM	82.78	17.22
sc_console	TGM	75.73	24.27
sc_web_browsing	TGM	92.42	7.58
sc_map	TGM	75.16	24.84
sc_programming	TGM	91.69	8.31
sc_SlideShow	TGM	97.77	2.23
sc_robot	A	95.65	4.35
EBURainFruits	CC	99.86	0.14
Kimono1	CC	94.71	5.29
Average (M+TGM)		85.56	14.44
Average (A+CC)		96.74	3.26
Average (All)		87.95	12.05

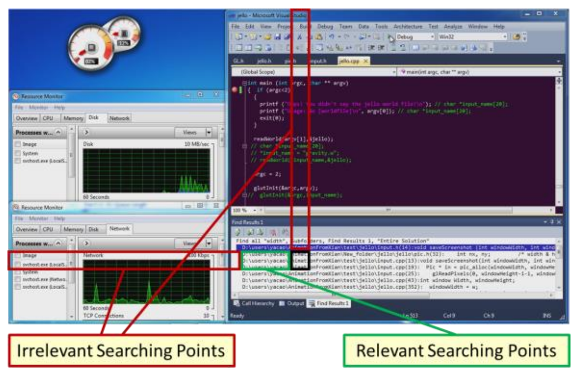
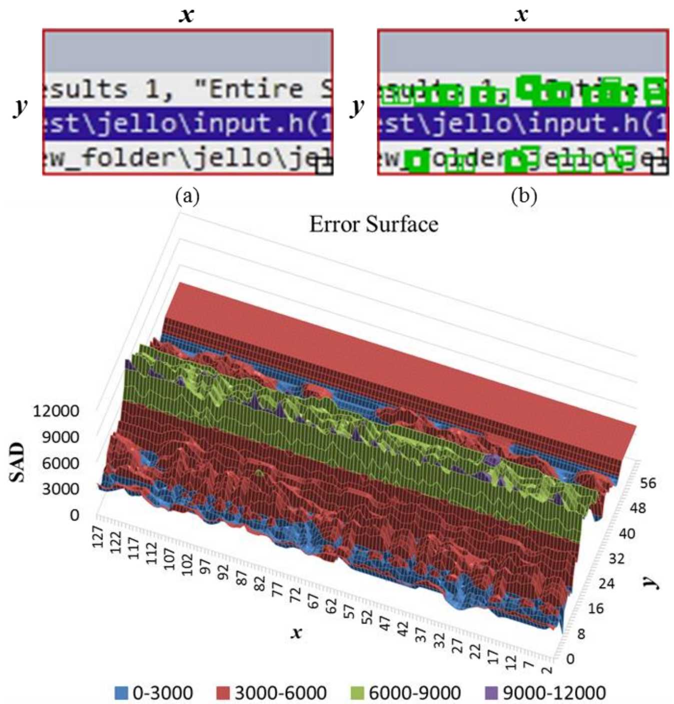


Fig. 4 Illustration of idea for hash based pre-defined area search.



(c)

Fig. 5 An example of hash based pre-defined area search, (a) the conventional search window, (b) candidates with the same hash value as the CU to be searched (green), and (c) the corresponding SAD error surface.

Fig. 1 used by pre-defined area search are fewer in which the resulted BV is always shorter. It is noted that the shorter the distance between the PU candidate and the current PU, the higher the tolerance of the distortion can be accepted. In other words, we can have larger SAD when the BV cost is low, as shown in (2). Hence, new formats of the hash values for different PU sizes with relaxed constraint is desired to include more search points with larger SAD. Although the hash value with relaxed constraint induces fewer reduction in search points, it increases the chance to find the best IntraBC cost when both SAD and BV cost are taken into account. Our proposed hash value used for speeding up the pre-defined area search for 8×8 CU with 2*N*×2*N* PU is:

$$Hash_{2N\times 2N} = (MSB_m(DC_0) \ll 3m) + (MSB_m(DC_1) \ll 2m) + (MSB_m(DC_2) \ll m) + MSB_m(DC_3) \quad (11)$$

For 8×8 CU with 2*N*×*N* PU, the hash value is:

$$Hash_{2N\times N} = (MSB_m(DC_k) \ll m) + MSB_m(DC_{k+2}) \quad (12)$$

where k can be 0 or 1. Similarly, for 8×8 CU with *N*×2*N* PU, the hash value is:

$$Hash_{N\times 2N} = (MSB_m(DC_k) \ll m) + MSB_m(DC_{k+1}) \quad (13)$$

where k can be 0 or 2. The intermediate values DC_k estimated in (6) are reused for estimating our proposed hash values in (11) to (13). With (11) to (13), the new hash values are adopted to the pre-defined area search by checking the hash values of PU candidates with different PU sizes. It is noted that there is no hash based fast pre-defined area search for 16×16 CU as only 2*N*×2*N* PU is supported, which occupies much less encoding time compared with 8×8 CU that supports various PU sizes. To relax the constraint of the hash value as discussed above, gradient is not included in (11) to (13) so that more PU candidates can be included for the IntraBC cost estimated by (2). It is important especially when the PU candidates have a low BV Cost. Fig. 5 shows an example of our proposed hash based pre-defined area search. Conventionally, all search points in Fig. 5(a) are necessary to be searched within the search window. With the proposed approach, only the candidates (green) in Fig. 5(b) with the same hash value as the current CU are searched. As can be seen in Fig. 5(c), the candidates usually are at the points with small values of SAD. Detailed analysis for selecting the values of *m* is discussed in Section IV.A.

IV. Experimental Results

To evaluate the performance of the proposed techniques, we have performed simulations using the HEVC reference software SCM-6.0 [43-44]. YUV 4:4:4 sequences, *Basketball_Screen*, *MissionControlClip2*, *MissionControl-Clip3*, *ChineseEditing*, *sc_flyingGraphics*, *sc_desktop*, *sc_console*, *sc_web_browsing*, *sc_map*, *sc_programming*, *sc_SlideShow*, *sc_robot*, *EBURainFruits* and *Kimono1*, which are recommended by the JCT-VC [47], were encoded. The coding conditions followed the Common Test Conditions (CTC) for SCC specified in [47]. The experiments were conducted on the Dell Precision T1700 computer with an Intel i7-4770 3.40GHz processor and 16GB memory. Comprehensive comparison has been done with the combination of various approaches listed in the following:

- HEVC: Conventional HEVC SCC extension but with both Palette and IntraBC modes disabled.
- HEVC+SCC: Conventional HEVC SCC extension with both Palette and IntraBC modes.
- HEVC+SCC (Palette Only): Conventional HEVC SCC extension with the use of Palette mode only.
- HEVC+SCC+(A): HEVC+SCC plus skipping IntraBC and Palette modes with early CU pruning based on pixel exactness (Section III A).
- HEVC+SCC+(A)+(B): HEVC+SCC+(A) plus early CU splitting based on partial 2D pixel exactness (Section III B).
- HEVC+SCC+(A)+(B)+(C): HEVC+SCC+(A)+(B) plus early CU pruning and early termination based on RD cost (Section III C).
- HEVC+SCC+(A)+(B)+(C)+(D): HEVC+SCC+(A) +(B)+(C) plus hash based pre-defined area search for IntraBC mode (Section III D).

It is noted that the fast approaches in [29-30] and the hash search in [31] have been adopted in HEVC+SCC, which are already mentioned in Section II. Hence, the HEVC+SCC used for comparison is already a fast SCC encoding scheme with the use of fast approaches in [29-31].

A. Determination of Parameters

For HEVC+SCC+(A)+(B), early CU splitting is decided if there are X_{CU_SIZE} sub-CUs in the current CU with the property of 2D pixel exactness, i.e. partial 2D pixel exactness. As discussed in Section III.B, X_{CU_SIZE} depends on the size of the current CU, *CU_SIZE*, since SCC mode decision for different *CU_SIZEs* goes through different processes, as shown in Fig. 2.

TABLE VI
BDBR (%) AND ENCODING TIME (%) FOR HEVC+SCC+(A)+(B) USING DIFFERENT VALUES OF X_{CU_SIZE} AGAINST HEVC+SCC [29-31]

Average (All)	$X_{64\times64} = 3$	$X_{64\times64} = 2$	$X_{64\times64} = 1$
BDBR	+0.14	+0.15	+0.14
Encoding Time	-6.36	-6.84	-7.10
Average (All)	$X_{32\times32} = 3$	$X_{32\times32} = 2$	$X_{32\times32} = 1$
BDBR	+0.16	+0.34	+0.45
Encoding Time	-6.03	-7.08	-8.65
Average (All)	$X_{16\times16} = 3$	$X_{16\times16} = 2$	$X_{16\times16} = 1$
BDBR	+0.53	+0.90	+1.19
Encoding Time	-9.06	-10.80	-12.64

TABLE VII
BDBR (%) AND ENCODING TIME (%) FOR HEVC+SCC+(A)+(B)+(C) USING DIFFERENT VALUES OF N AGAINST HEVC+SCC [29-31]

Average (All)	$n = 16$	$n = 32$	$n = 48$	$n = 64$
BDBR	+0.78	+0.86	+1.43	+2.79
Encoding Time	-11.95	-14.47	-15.58	-18.39

TABLE VIII
BDBR (%) AND ENCODING TIME (%), AND SEARCH POINT REDUCTION (%) FOR HEVC+SCC+(A)+(B)+(C)+(D) USING DIFFERENT VALUES OF M AGAINST HEVC+SCC [29-31]

Average (All)	$m = 0$	$m = 1$	$m = 2$	$m = 3$
BDBR	+0.86	+1.26	+1.52	+1.88
Encoding Time	-14.47	-15.56	-17.07	-17.44
Number of Search Points	-15.24	-47.10	-64.24	-75.35

Table VI tabulates the average Bjontegaard Delta Bitrate (BDBR) [48] and encoding time reduction against HEVC+SCC in percentage for HEVC+SCC+(A)+(B) using different values of X_{CU_SIZE} to define the properties of partial 2D pixel exactness for 64×64, 32×32 and 16×16 CUs.

From Table VI, when early CU splitting is enabled only at 64×64 CU, the largest encoding time reduction of 7.10% can be achieved at $X_{64\times64}$=1, while BDBR increases for both cases are very similar. Thus, $X_{64\times64}$ is set to 1. A similar situation happens when early CU splitting is also enabled at 32×32 CU with $X_{32\times32}$=1. Consequently, $X_{32\times32}$ is also set to 1. On the contrary, for 16×16 CU, a significant BDBR increases of 0.90% and 1.19% when $X_{16\times16}$ is set to 1 and 2, respectively. But there is a limited BDBR increase of 0.53% while still having the encoding time reduction of 9.06% when $X_{16\times16}$ is set to 1. As a result, the choice of {$X_{64\times64}$=1, $X_{32\times32}$=1, $X_{16\times16}$=3} is selected in the early CU splitting decision and it also verifies the justification explained in Section III.B that a 64×64 or 32×32 CU uses a smaller value of X_{CU_SIZE}, while an 16×16 CU employs a larger value of X_{CU_SIZE} for defining the property of partial 2D pixel exactness.

For HEVC+SCC+(A)+(B)+(C), i.e. the approach of early CU pruning and early termination based on RD cost, different values of n for computing the threshold TH_4 as in (10) were tested. Results are tabulated in Table VII. With n = 48 and 64, the BDBR increases largely though there is larger encoding time reduction. Thus, we select n = 32 for estimating TH_4, which has a limited BDBR impact of only 0.86% bitrate increase with 14.47% encoding time reduction against HEVC+SCC.

Finally, Table VIII tabulates the average BDBR, encoding time, and search point reduction in percentage (%) for HEVC+SCC+(A)+(B)+(C)+(D) using different values of m in (11) to (13). m = 0 actually means HEVC+SCC+(A)+(B)+(C). The value of m indicates the degree of constraint being relaxed as discussed in Section III.C. The smaller the value of m, the more the search points to be tested since the hash value becomes less tight. The value of m can be considered a tradeoff to find the best IBC cost and the number of search points being tested. From Table VIII, it can be seen that even with m = 3, our proposed scheme still can obtain less than 2% increase in BDBR. We chose the case of m = 2 as optimum since it can have better BDBR while still having similar encoding time reduction compared with the case of m = 3. Keeping m=2 has additional benefit by restricting the hash value in (11) to one byte that can save the memory requirement.

Table VII and Table VIII also shows that our early CU pruning and early termination based on RD cost and hash based pre-defined area search are computational scalable by choosing different values of n and m. After massive experimental testing, it achieves the best operation point when n and m are set to 32 and 3, respectively, which are adopted in the following simulations.

B. Comparison with HEVC+SCC

Table IX and Table X tabulate the BDBR and encoding time against HEVC+SCC in percentage for each testing sequence, respectively. As can be seen in these tables, only about 0.14% increase in BDBR on average for HEVC+SCC+(A) compared with HEVC+SCC, while the encoding time is reduced by about 6.34%. The reason for increased bitrate is that if both the SAD and BV cost in (2) tends to zero for IntraBC mode, as mentioned in Section III.A, the overhead of the Intra mode, i.e. the luminance and chrominance prediction directions, is a bit larger than that of the IntraBC mode, i.e. the skip mode index or the BV difference.

For HEVC+SCC+(A)+(B), it sacrifices about 0.53% BDBR on average compared with HEVC+SCC, but the encoding time is remarkably reduced by about 9.06% since early CU splitting is done, which can skip a substantial number of RD cost estimation in (1). Besides, the complexity induced by IntraBC in (2) can also be reduced though the exhaustive block matching for IntraBC mode is only done for 16×16 and 8×8 CUs. The increase in bitrate is due to more CU overheads are required to be encoded as the number of CU partition increases. Furthermore, about 0.86% increase in BDBR on average for HEVC+SCC+(A)+(B)+(C) with 14.47% time reduction compared with HEVC+SCC. The proposed CU pruning technique reduces the number of CU candidates, which implies that the complexities induced by the RD cost estimation in (1) and IntraBC mode in (2) can be reduced significantly.

Lastly, the encoding reduction of 17.07% for HEVC+SCC +(A)+(B)+(C)+(D) compared with HEVC+SCC trade off against 1.52% increase in the BDBR on average. By checking the hash values of PU candidates and the current PU with the removal of the gradient term, *Grad*, in (6), which becomes our hash functions from (11) to (13), irrelevant searching areas, as shown in Fig. 4, can still be skipped such that the complexity induced by (2) can be reduced as resulted in Table XI. With the gradient term, BDBR is increased to 1.95% with 18.01% average time reduction which is only about 1% further time reduction. It is because the time saved in block matching has been penalized by the time for estimating the gradient term.

TABLE IX
BDBR (%) OF VARIOUS TECHNIQUES AGAINST HEVC+SCC [29-31]

Sequences	HEVC +SCC (Palette Only)	HEVC +SCC +(A)	HEVC +SCC +(A)+(B)	HEVC +SCC +(A)+(B) +(C)	HEVC +SCC +(A)+(B) +(C)+(D)
Basketball_Screen	+42.36	+0.28	+0.64	+0.71	+1.38
MissionControlClip2	+46.86	+0.08	+0.25	+0.33	+1.00
MissionControlClip3	+81.10	+0.06	+0.75	+1.05	+2.20
ChineseEditing	+24.32	+0.00	+0.31	+0.47	+0.89
sc_flyingGraphics	+67.83	+0.28	+0.60	+1.14	+3.41
sc_desktop	+131.28	+0.35	+0.70	+1.66	+2.74
sc_console	+35.94	+0.38	+1.08	+2.73	+3.40
sc_web_browsing	+116.11	-0.12	+0.89	+1.30	+1.82
sc_map	+7.17	+0.11	+0.36	+0.25	+0.78
sc_programming	+41.10	+0.21	+0.47	+0.72	+1.43
sc_SlideShow	+9.73	+0.32	+1.31	+1.61	+2.01
sc_robot	+1.64	+0.02	+0.02	+0.01	+0.17
EBURainFruits	+0.11	+0.00	+0.00	+0.00	+0.07
Kimono1	+0.00	+0.00	+0.00	+0.00	+0.03
Average (M+TGM)	+54.89	+0.18	+0.67	+1.09	+1.91
Average (A+CC)	+0.59	+0.01	+0.01	+0.01	+0.09
Average (All)	+43.25	+0.14	+0.53	+0.86	+1.52

TABLE X
ENCODING TIME (%) OF VARIOUS TECHNIQUES AGAINST HEVC+SCC [29-31]

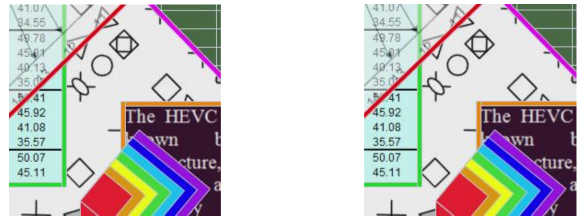
Sequences	HEVC +SCC (Palette Only)	HEVC +SCC +(A)	HEVC +SCC +(A)+(B)	HEVC +SCC +(A)+(B) +(C)	HEVC +SCC +(A)+(B) +(C)+(D)
Basketball_Screen	-29.31	-7.22	-9.36	-14.82	-17.94
MissionControlClip2	-28.61	-4.27	-5.65	-13.72	-16.35
MissionControlClip3	-29.86	-3.43	-5.65	-11.55	-14.89
ChineseEditing	-37.93	-5.40	-7.85	-12.23	-16.28
sc_flyingGraphics	-32.43	-3.80	-7.52	-10.67	-13.52
sc_desktop	-29.53	-8.46	-14.95	-19.10	-20.58
sc_console	-24.70	-7.50	-10.53	-15.52	-16.39
sc_web_browsing	-26.15	-4.41	-15.88	-19.38	-20.86
sc_map	-43.68	-5.61	-6.11	-10.11	-13.19
sc_programming	-24.32	-6.50	-9.02	-13.37	-16.11
sc_SlideShow	-20.31	-31.95	-34.86	-41.96	-43.45
sc_robot	-33.20	-0.48	-0.21	-5.99	-9.49
EBURainFruits	-36.74	+0.26	+0.48	-9.54	-13.32
Kimono1	-29.81	+0.02	+0.33	-4.56	-6.57
Average (M+TGM)	-29.71	-8.05	-11.58	-16.58	-19.05
Average (A+CC)	-33.25	-0.07	+0.20	-6.70	-9.80
Average (All)	-30.47	-6.34	-9.06	-14.47	-17.07

TABLE XI
BDBR (%) AND ENCODING TIME (%) FOR HEVC+SCC+(A)+(B)+(C)+(D) WITH AND WITHOUT GRADIENT AGAINST HEVC+SCC [29-31]

Average (All)	Hash value without gradient	Hash value with gradient
BDBR	+1.52	+1.95
Encoding Time	-17.07	-18.01

Regarding the better BDBR of the new hash values, the search points filtered by hash functions with the gradient term are the subset of those without the gradient term. The new hash values with relaxed constraint in (11) to (13) can include more search points resulting in high possibility of getting better IntraBC cost in (2), as mentioned in Section III.D. Thereby, the proposed algorithm adopts the hash functions without the gradient term as in (11) to (13).

Subjective quality for the *sc_flyingGraphics* sequence, which has diverse screen content and the largest BDBR increase, as shown in Table IX, is also illustrated in Fig. 6. We can see that there is no obvious subjective quality impact between HEVC+SCC and HEVC+SCC+(A)+(B)+(C)+(D).



(a) (b)

Fig. 6 Subjective quality of *sc_flyingGraphics* with QP 27 using (a) HEVC+SCC (45.09dB), and (b) HEVC+SCC+(A)+(B)+(C)+(D) (45.08dB).

TABLE XII
NUMBER OF CUS (%) THAT CHECKED THE INTRA MODE, PALETTE MODE, AND INTRABC MODE, AS WELL AS SEARCH POINT VARIATION (%) FOR PROPOSED TECHNIQUES AGAINST HEVC+SCC [29-31]

Sequences	HEVC +SCC +(A)	HEVC +SCC +(A)+(B)	HEVC +SCC+(A) +(B)+(C)	HEVC +SCC+(A) +(B)+(C)+(D)
Number of CUs (%) that checked the Intra mode				
Average (M+TGM)	-14.61	-16.96	-21.17	-21.05
Average (A+CC)	-0.01	-0.01	-4.53	-4.53
Average (All)	-11.48	-13.33	-17.60	-17.51
Number of CUs (%) that checked the Palette mode				
Average (M+TGM)	-24.02	-25.73	-29.46	-29.40
Average (A+CC)	-0.03	-0.03	-4.59	-4.58
Average (All)	-18.88	-20.23	-24.13	-24.08
Number of CUs (%) that checked the IntraBC mode				
Average (M+TGM)	-25.01	-25.61	-42.65	-42.62
Average (A+CC)	-0.03	-0.03	-28.80	-28.80
Average (All)	-19.66	-20.13	-39.68	-39.66
Search point variation (%)				
Average (M+TGM)	-4.23	-4.26	-14.54	-65.18
Average (A+CC)	+0.01	+0.01	-17.81	-60.77
Average (All)	-3.32	-3.35	-15.24	-64.24

C. *Complexity Analysis of Proposed Techniques*

Table XII tabulates the number of CUs in percentage that perform the checking of Intra, Palette and IntraBC modes during the mode decision process, as well as the search point variation in percentage for IntraBC of various techniques against HEVC+SCC. A decrease in number of mode checking implies time saving for searching and RD cost estimation within that mode. A decrease in number of search points indicates the time saving for IntraBC cost estimation in (2).

Our proposed HEVC+SCC+(A) is effective in the sense that the numbers of CUs that check the Intra, Palette and IntraBC modes are reduced by 11.48%, 18.88% and 19.66%, respectively, and the number of search points is reduced by 3.32% compared with HEVC+SCC.

For HEVC+SCC+(A)+(B), the reduction in search points is still maintained at around 3.35%, since IntraBC block matching is only done for 16×16 and 8×8 CUs while early CU splitting is usually performed for large CU sizes of 64×64 and 32×32. But the numbers of CUs that check the Intra, Palette and IntraBC modes are reduced by 13.33%, 20.23% and 20.13%, respectively.

For HEVC+SCC+(A)+(B)+(C), the numbers of CUs that check the Intra, Palette and IntraBC modes are reduced by 17.60%, 24.13% and 39.68%, respectively, and the number of search points is also reduced by 15.24%. It is due to the reason that our proposed early CU pruning can skip the mode decision process and the number of search points within IntraBC mode for smaller CUs. And our proposed early termination can skip IntraBC mode checking when the RD cost for either Intra mode or Palette mode is small.

Likewise, for HEVC+SCC+(A)+(B)+(C)+(D), the numbers of CUs that check Intra, Palette and IntraBC modes achieve similar reduction as HEVC+SCC+(A)+(B)+(C), since the proposed hash based pre-defined area search mainly focuses on reducing the complexity induced by (2) within IntraBC mode only. It is expected that the number of search points is reduced significantly by 64.24%. This means that indeed there are many redundant search points for IntraBC in HEVC+SCC that make the encoder complexity impractically high.

D. *Integration with State-of-the-art SCC algorithms*

We also evaluated the performances of our proposed algorithm incorporated with the state-of-the-art approaches [40-42]. It is noted that one-to-one comparison between our proposed algorithm and the algorithms in [40-42] is not straightforward, since different algorithms contribute to different aspects of SCC. Our proposed algorithm mainly handles the screen content CUs (SCCUs) and is self-contained, while the algorithms in [40-42] might use CU depth prediction that requires other frame or a classifier for handling SCCUs and camera-captured content CUs (CCCUs) separately. In this section, we will show that the proposed algorithm is an excellent complement of the approaches in [40-42] to form a very efficient SCC encoder.

In this section, our proposed HEVC+SCC+(A)+(B)+(C)+(D) is denoted as OURS for simplicity. Table XIII shows the performances of OURS incorporated with the algorithms in [40-42]. For the sake of completeness and discussion, the performances of OURS, Zhang [40], Duanmu [41] and Lei [42] are also listed. Notice that the SCM versions used in [40-42] are from 2.0 to 4.0 while ours is 6.0. There are numerous enhancements, speed-up techniques and codes clean-up related to IntraBC mode in SCM-6.0 compared with the older versions. In the older versions, the BV signal in IntraBC mode was not unified with the inter mode, which only has left and above BVs as predictors with no skip and merge modes. Consequently, there was also no condition checking whether skip mode is the best mode before going into the time-consuming IntraBC block matching, as shown in Fig. 2. Moreover, $N \times N$ IntraBC search was done after $2N \times N$ search. We then implemented the approaches in [40-42] in SCM-6.0 for fair comparison.

1) *Integration with [40] and its Performance Comparison*

Both of OURS and Zhang [40] are specifically designed for complexity reduction on SCCUs. The evidence can be seen in Table XIII in which the BDBR and average time reduction of OURS for (A+CC) sequences, which mainly consists of CCCUs or almost all CUs are encoded by the conventional Intra mode, are only 0.09% and 9.80%, respectively. The same observation can be found in Zhang [40], since it also focuses on the same thing as OURS but mainly for stationary CUs.

TABLE XIII
BDBR (%) AND ENCODING TIME (%) FOR DIFFERENT APPROACHES AGAINST HEVC+SCC [29-31]

Approaches (SCM-6.0)	Average (M+TGM)		Average (A+CC)		Average (All)	
	BDBR	Time	BDBR	Time	BDBR	Time
OURS	+1.91	-19.05	+0.09	-9.80	+1.52	-17.07
Zhang [40]	+1.43	-39.47	+0.58	-9.52	+1.25	-33.05
Duanmu [41]	+1.50	-26.60	+0.90	-27.79	+1.37	-26.86
Lei [42]	+2.05	-27.32	+2.83	-55.96	+2.22	-33.45
OURS + Zhang's [40]	+3.96	-49.27	+0.71	-16.49	+3.26	-42.24
OURS + Duanmu [41]	+4.03	-39.51	+1.04	-37.10	+3.39	-39.00
OURS + Lei [42]	+3.91	-35.72	+2.97	-59.52	+3.71	-40.82

TABLE XIV
BDBR (%) AND ENCODING TIME (%) FOR OURS, ZHANG[40], AND THEIR INTEGRATION AGAINST HEVC+SCC [29-31] FOR ORIGINAL AND FRAME-SKIPPED SEQUENCES. NOTED THAT ONLY SOME TGM SEQUENCES ARE SHOWN.

Sequences	Zhang [40]		OURS		OURS + Zhang [40]	
	BDBR	Time	BDBR	Time	BDBR	Time
Non-Frame-Skipped Condition						
sc_flyingGraphics	+0.91	-3.59	+3.41	-13.52	+5.32	-18.40
sc_desktop	+2.30	-47.54	+2.74	-20.58	+5.79	-57.71
sc_console	+3.09	-37.47	+3.40	-16.39	+6.89	-50.70
sc_web_browsing	+1.90	-51.88	+1.82	-20.86	+4.45	-57.19
sc_map	+0.79	-36.94	+0.78	-13.19	+2.06	-45.77
sc_programming	+1.15	-40.71	+1.43	-16.11	+3.05	-48.74
sc_SlideShow	+1.14	-44.40	+2.01	-43.45	+3.89	-59.11
Average (M+TGM)	+1.43	-39.47	+1.91	-19.05	+3.96	-49.27
Average (A+CC)	+0.58	-9.52	+0.09	-9.80	+0.71	-16.49
Average (Overall)	+1.25	-33.05	+1.52	-17.07	+3.26	-42.24
Frame-Skipped Condition						
sc_flyingGraphics	+0.43	+0.01	+5.00	-11.64	+6.48	-13.76
sc_desktop	+1.96	-22.90	+3.41	-17.84	+6.61	-38.95
sc_console	+1.22	-3.18	+3.92	-16.40	+6.11	-21.78
sc_web_browsing	+0.69	-13.12	+3.28	-22.58	+5.16	-29.73
sc_map	+0.64	-18.30	+1.93	-18.68	+3.31	-34.50
sc_programming	+0.66	-12.64	+2.08	-14.92	+3.60	-26.97
sc_SlideShow	+1.37	-27.17	+2.02	-42.01	+3.71	-50.60
Average (M+TGM)	+0.91	-16.99	+2.54	-18.56	+4.26	-32.13
Average (A+CC)	+0.11	-5.13	+0.22	-9.80	+0.43	-13.84
Average (Overall)	+0.74	-14.45	+2.04	-16.68	+3.44	-28.21

From Table XIII, Zhang [40] provides better BDBR and larger time reduction on overall average. It is expected since Zhang [40] heavily relies on stationary region detection in the current frame and then reuses the CU depth information from the stationary regions in the previous frame. Then, it works well for stationary CUs, but additional memory is required for storing all pixels, CU depth and mode information of the previous frame. It is noted that there is no problem for using other frame information at the encoder but the modification of frame buffer management is still needed. Conversely, OURS is self-contained without the help of other frames and works for CUs with object movements or non-stationary CUs. Table XIV lists the BDBR and average time reduction of some TGM sequences for further comparison between OURS and Zhang [40]. For the sequences with rapid windows rotation and color changing such as *sc_flyingGraphics*, the time reduction of Zhang [40] diminishes to 3.59%, as shown in Table XIV, while the time reduction of OURS can achieve 13.52%. When the inter mode is enabled in the SCC encoder, the number of non-stationary CUs is reduced significantly. We also simulated this situation by using frame-skipped sequences as shown in Table XIV. The frame-skipped sequences are the sequences in CTC [47] with the first frame of each second encoded when inter frames are inserted in between. In Table XIV, the average time reduction of Zhang [40] for sequences (M+TGM) diminishes from 39.47% without frame-skipping to 16.99% for frame-skipped sequences. For *sc_flyingGraphics*, the time saving completely disappears. It shows that Zhang [40] has little impact on time reduction for sequences with high motion activity. In contrast, the average time reduction of OURS keeps nearly constant, around 19.05% and 18.56% for sequences (M+TGM) with and without frame-skipping, respectively. It verifies that Zhang [40] heavily relies on the number of stationary CUs in the coded sequence. Since OURS does not use information from other frames, its performance is insensitive to the motion activity in the sequence.

When the encoder is allowed to access the information from other frames, OURS can be integrated with Zhang [40] such that the temporal CU depth correlation can be utilized for stationary CUs. With the incorporation of Zhang [40] into OURS, OURS + Zhang [40] in Table XIV, 32.13% and 49.27% average time reduction can be obtained for sequences (M+TGM) with and without frame-skipping, respectively, which achieves more complexity savings than both OURS and Zhang [40] alone. This shows that different algorithms contribute to different aspects of SCC. Zhang [40] can speed up stationary CUs while OURS can work well for CUs with object movements. This is a good complement between OURS and Zhang [40].

2) Integration with [41-42] and its Performance Comparison

Unlike OURS and Zhang [40], which mainly consider how to speed up the SCC tools only, Duanmu [41] and Lei [42] divide CUs into CCCUs and SCCUs using a learning technique or some features in video content, and then adopt some conventional techniques for CCCUs. For instance, there are two classifiers in Duanmu [41], the partition classifier and the directional block classifier that are used to have fast CU partitioning and fast conventional Intra mode decision, for CCCUs, respectively. Also, there is a fast CU termination process for both SCCUs and CCCUs, named as Rate-Based Fast Termination (RBFT) in which its formulation, as mentioned by [41], is similar to the technique used in fast CU termination in HEVC [49]. For Lei [42], fast mode decision for CCCUs is applied while the CU depth prediction based on the neighboring and collocated CUs is adopted in both SCCUs and CCCUs [50]. These well-established HEVC techniques are applied to CCCUs in SCC, and some of these are further extended to SCCUs in SCC in [41-42]. However, OURS and Zhang [40] are much focused on SCCU complexity reduction techniques. Due to this reason, results in Table XIII show that Duanmu [41] and Lei [42] obtained larger increase in BDBR and average time reduction for (A+CC) sequences because they use the above techniques for CCCUs. That is also the reason why the BDBR or average time reduction of OURS for (M+TGM) sequences might not be better than those in Duanmu [41] and Lei [42], since there are also CCCUs in (M+TGM) sequences.

With only the Duanmu [41], 26.86% average time reduction and 1.37% increase in BDBR are obtained. From the above reasons, it is difficult to evaluate the performances of OURS and Duanmu [41] without isolating the conventional HEVC techniques in [41]. As mentioned in [41], its framework allows a new algorithm on each individual SCC tool for additional encoder speed-up. It is more interesting to show how well our proposed algorithm worked with Duanmu [41]. With the incorporation of Duanmu [41] into OURS, i.e. OURS + Duanmu [41], larger average time reduction of 39.00% is obtained with 3.39% increase in BDBR only. It means that it can further reduce the complexity of Duanmu [41].

For Lei [42], one of their proposed techniques is the CU depth prediction based on the neighboring and collocated CUs, which is the work in [50] for the conventional HEVC. In the results shown in [42], the CU depth prediction contributes the major portion to the overall time saving in Lei [42]. On the other hand, OURS is self-contained, and only uses the information within the CU. With only the Lei [42], 33.45% average time reduction and 2.22% increase in BDBR are obtained. When the access to the information from other frames is feasible, OURS can be integrated with the CU depth prediction in Lei [42] such that the temporal CU depth correlation can be utilized, i.e. OURS + Lei [42]. From Table XIII, OURS + Lei [42] can obtain larger average time reduction of 40.82% with BDBR increased by 3.71%.

From Table XIII, we can see that the BDBR is further increased when OURS is incorporated with [40-42]. It is reasonable because there is no fine-tuning for the algorithms during code merging process, which makes the fast techniques become greedy. Further encoding time reduction achieved by our integrations proves that the areas focused by OURS are different from the works in [40-42]. These integrated frameworks could be a point for our immediate future work.

V. Conclusions

Intra Block Copy (IntraBC) mode helps to increase the coding efficiency of SCC by finding the repeating patterns within the same frame using pre-defined area search and hash search, but also makes the encoding complexity impractically high. Hence, we propose to skip IntraBC mode based on the pixel exactness. Besides, unnecessary checking of IntraBC mode is skipped by early CU splitting and pruning as well as early termination. Furthermore, checking of hash values between PU candidates and the current PU is carried out before SAD cost estimation within IntraBC mode. By our fast decision techniques on SCC intra modes, the encoding time can be reduced by 17.07% on average, while there is only 1.52% increase in BDBR compared with the conventional HEVC SCC extension. Our proposed algorithm is mainly designed for coding the screen content CUs. Results also demonstrated that it can work in conjunction with other recent frameworks for a remarkable complexity reduction.

VI. Acknowledgement

We would like to thank the authors from [40-41] for the provision of source codes.

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, Feb. 2011.

[2] G. J. Sullvian, 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.

[3] "Joint Call for Proposals for Coding of Screen Content," *ISO/IEC JTC1/SC29/WG11*, document N14175, San Jose, U.S.A., Jan. 2014.

[4] "Requirements for Future Extensions of HEVC in Coding Screen Content," *ISO/IEC JTC1/SC29/WG11*, document N14174, San Jose, U.S.A., Jan. 2014.

[5] G. J. Sullivan, J. M. Boyce, Y. Chen; J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized Extensions of High Efficiency Video Coding (HEVC)," *IEEE J. Sel. Topics. Signal Process*, vol. 7, no. 6, pp.1001-1016, Dec. 2013.

[6] 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. 2015.

[7] T. Lin, P. Zhang, S. Wang, K. Zhou, and X. Chen, "Mixed Chroma Sampling-Rate High Efficiency Video Coding for Full-Chroma Screen Content," *IEEE Trans. Circuits Syst. Video Technol.*, vol.23, no.1, pp.173-185, Jan. 2013.

[8] S. Wang, T. Lin, K. Zhou, P. Zhang, and X. Chen, "Pseudo-2D-Matching Based Enhancement to High Efficiency Video Coding for Screen Contents," *Springer J. Multimedia Tools and Appl.*, vol. 74, no. 18, pp.7754-7771, Sep. 2015.

[9] Z. Pan, H. Shen, Y. Lu, S. Li, and N. Yu, "A Low Complexity Screen Compression Scheme for Interactive Screen Sharing," *IEEE Trans. Circuits Syst. Video Technol.*, vol.23, no.6, pp.949-960, Jun. 2013.

[10] S. Wang, K. Gu, S. Ma, and W. Gao, "Joint Chroma Downsampling and Upsampling for Screen Content Image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1595-1609, Sep. 2015.

[11] 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. of Digital Image Process.*, pp.888-892, Hong Kong, China, Aug. 2014.

[12] 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.

[13] Y. Piao, J.H. Min, and J. Chen, "Encoder Improvement of Unified Intra Prediction," *JCT-VC*, JCTVC-C207, Guangzhou, China, Oct. 2010.

[14] Weijia Zhu, Wenpeng Ding, Jizheng Xu, Yunhui Shi, and Baocai Yin, "Screen Content Coding Based on HEVC Framework," *IEEE Trans. Multimedia*, vol.16, no.5, pp.1316-1326, Aug. 2014.

[15] 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.

[16] Y.-C. Sun, T.-D. Chuang, P. Lin, Y.-W. Chen, S. Liu, and Y.-W. Huang, *et al.*, "Palette Mode – A New Coding Tools in Screen Content Coding Extensions of HEVC," in *Proc. of IEEE Int. Conf. on Image Process.*, pp. 2409-2413, Quebec City, Canada, Sep. 2015.

[17] X. Xiu, Y. He, R. Joshi, M. Karczewicz, P. Onno, and C. Gisquet, *et al.*, "Palette-Based Coding in the Screen Content Coding Extension of the HEVC Standard," in *Proc. of Data Compression Conf.*, pp.253-262, Snowbird, U.S.A., Apr. 2015.

[18] X. Zhang, R. A. Cohen, and A. Vetro, "Independent Uniform Prediction Mode for Screen Content Video Coding," in *Proc. of IEEE Int. Conf. on Visual Comm. and Image Process.*, pp.129-132, Valletta, Malta, Dec. 2014.

[19] M. Zhang, Y. Guo, and H. Bai, "Fast Intra Partition Algorithm for HEVC Screen Content Coding," in *Proc. of IEEE Int. Conf. on Visual Comm. and Image Process.*, pp.390-393, Valletta, Malta, Dec. 2014.

[20] S.-H. Tsang, Y.-L. Chan, and W.-C. Siu, "Fast and Efficient Intra Coding Technique for Smooth Regions in Screen Content Coding Based on Boundary Prediction Samples," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, pp.1409-1413, Brisbane, Australia, Apr. 2015.

[21] T. Lin, X. Chen, and S. Wang, "Pseudo-2D-Matching Based Dual-Coder Architecture for Screen Contents Coding," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, pp.1-4, San Jose, U.S.A., Jul. 2013.

[22] T. Lin, K. Zhou, X. Chen, and S. Wang, "Arbitrary Shape Matching for Screen Content Coding," in *Proc. of Picture Coding Symp.*, pp.369-372, San Jose, U.S.A., Dec. 2013.

[23] B. Li, J. Xu, and F. Wu, "1-D Dictionary Mode for Screen Content Coding," in *Proc. of IEEE Int. Conf. on Visual Comm. and Image Process.*, pp.189-192, Valletta, Malta, Dec. 2014.

[24] F. Zou, Y. Chen, M. Karczewicz, and V. Seregin, "Hash Based Intra String Copy for HEVC Based Screen Content Coding," in *Proc. of IEEE Int. Conf. on Multimedia & Expo*, pp.1-4, Torino, Italy, Jun.-Jul. 2015.

[25] K. Zhou, L. Zhao, and T. Lin, "A Hardware Decoder Architecture for General String Matching Technique," *IEEE J. Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 560-572, Dec. 2016.

[26] C.-C. Chen, X. Xu, R.-L. Liao, W.-H. Peng, S. Liu, and S. Lei, "Screen Content Coding Using Non-Square Intra Block Copy for HEVC," in *Proc. of IEEE Int. Conf. on Multimedia and Expo*, pp.1-6, Chengdu, China, Jul. 2014.

[27] M. Zhang, Y. Zhang, X. Peng, and J. Xu, "Screen Content Coding for HEVC by Improved Line-Based Intra Block Copy," in *Proc. of IEEE Int. Conf. on Image Process.*, pp.3744-3748, Paris, France, Oct. 2014.

[28] C.-C. Chen, and W.-H. Peng, "Intra Line Copy for HEVC Screen Content Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp.1568-1579, Mar. 2016.

[29] M. Budagavi, and D.-K. Kwon, "Intra Motion Compensation and Entropy Coding Improvements for HEVC Screen Content Coding," in *Proc. of Picture Coding Symp.*, pp.365-368, San Jose, U.S.A., Dec. 2013.

[30] M. Budagavi, and D.-K. Kwon, "Fast Intra Block Copy (IntraBC) Search for HEVC Screen Content Coding," in *Proc. of IEEE Int. Symp. on Circuits Syst.*, pp.9-12, Melbourne, Australia, Jun. 2014.

[31] X. Xu, S. Liu, T.-D. Chuang, Y.-W. Huang, S.-M. Lei, and K. Rapaka *et al.*, "Intra Block Copy in HEVC Screen Content Coding Extensions," *IEEE J. Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 409-419, Dec. 2016.

[32] W. Zhu, W. Ding, J. Xu, Y. Shi, and B. Yin, "2-D Dictionary Based Video Coding for Screen Contents," in *Proc. of Data Compression Conf.*, pp.43-52, Snowbird, U.S.A., Mar. 2014.

[33] W. Zhu, W. Ding, J. Xu, Y. Shi, and B. Yin, "Hash Based Block Matching for Screen Content Coding," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp.935-944, Jul. 2015.

[34] B. Li, J. Xu, and F. Wu, "A Unified Framework of Hash-Based Matching for Screen Content Coding," in *Proc. of IEEE Int. Conf. of Visual Comm. and Image Proc.*, pp.530-533, Valletta, Malta, Dec. 2014.

[35] X. Xu, S. Liu, J. Ye, and S. Lei, "PU Level Intra Block Copying with Flipping Mode," in *Proc. of Asia-Pacific Signal and Inf. Process. Association Annual Summit and Conf.*, pp.1-7, Siem Reap, Cambodia, Dec. 2014.

[36] K. Zhang, J. An, X. Zhang, H. Huang, and S. Lei, "Symmetric Intra Block Copy in Video Coding," in *Proc. of IEEE Int. Symp. on Circuits Syst.*, pp.521-524, Lisbon, Portugal, May 2015.

[37] J. Lainema, "Intra Block Copy Masking," in *Proc. of IEEE Int. Conf. on Consum. Electron.*, pp.333-336, Las Vegas, U.S.A., Jan. 2015.

[38] Z. Zhang, and V. Sze, "Rotate Intra Block Copy for Still Image Coding," in *Proc. of IEEE Int. Conf. on Image Process.*, Quebec City, Canada, pp. 4102-4106, Sep. 2015.

[39] X. Xu, S. Liu, T.-D. Chuang, and S. Lei, "Block Vector Prediction for Intra Block Copying in HEVC Screen Content Coding," in *Proc. of Data Compression Conf.*, pp.273-282, Snowbird, U.S.A., Apr. 2015.

[40] H. Zhang, Q. Zhou, N. Shi, F. Yang, X. Feng, and Z. Ma, "Fast Intra Mode Decision and Block Matching for HEVC Screen Content Compression," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, pp. 1377-1381, Shanghai, China, Mar. 2016.

[41] 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. Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 4, pp. 517-531, Dec. 2016.

[42] 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. Broadcasting*, vol. 63, no. 1, pp. 48-58, Mar. 2017.

[43] "Screen Content Coding Test Model 6 (SCM 6)," *JCT-VC*, JCTVC-V1014, Geneva, Switzerland, October 2015.

[44] HM-16.7+SCM-6.0, HEVC Test Model Version 16.7 Screen Content Model Version 6.0, [Online], Available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7+SCM-6.0/.

[45] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block Partitioning Structure in the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol.22, no.12, pp.1697-1706, Dec. 2012.

[46] 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. of Asia-Pacific Signal and Inf. Process. Association Annual Summit and Conf.*, pp. 396-400, Hong Kong, China, Dec. 2015.
[47] "Common Test Conditions for Screen Content Coding," *JCT-VC*, JCTVC-U1015, Warsaw, Poland, Jun. 2015.
[48] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD Curves," *VCEG*, VCEG-M33, Austin, U.S.A., Apr. 2001.
[49] S. G. Blasi, E. Peixoto, B. Macchiavello, E. M. Hung, I. Zupancic, and E. Izquierdo, "Context Adaptive Mode Sorting for Fast HEVC Mode Decision," in *Proc. of IEEE Int. Conf. Image Process.*, Quebec City, Canada, pp. 1478–1482, Sep. 2015.
[50] L. Shen, Z. Zhang, and Z. Liu, "Effective CU Size Decision for HEVC Intracoding," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232-4241, Oct. 2014.

Sik-Ho Tsang (M'10) received the Ph.D. degree from The Hong Kong Polytechnic University (PolyU), Hong Kong, in 2013.

He was a Postdoctoral Fellow from 2013 to 2016, and involved numerous industrial projects for video coding and transcoding. He is currently a Research Fellow in PolyU. He has authored numerous international journals, conferences and patents. His current research interests involve data science, machine learning and deep learning in video coding such as HEVC, Versatile Video Coding, multiview plus depth coding, screen content coding, light-field video coding, and 360-degree omnidirectional video coding.

He serves as a reviewer of international journals including the IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and Elsevier Journal of Signal Processing: Image Communication.

Yui-Lam Chan (S'94-A'97-M'00) received the B.Eng. (Hons.) and Ph.D. degrees from The Hong Kong Polytechnic University, Hong Kong, in 1993 and 1997, respectively.

He joined The Hong Kong Polytechnic University in 1997, where he is currently an Associate Professor with the Department of Electronic and Information Engineering. He is actively involved in professional activities. He has authored over 110 research papers in various international journals and conferences. His research interests include multimedia technologies, signal processing, image and video compression, video streaming, video transcoding, video conferencing, digital TV/HDTV, 3DTV/3DV, multiview video coding, machine learning for video coding, and future video coding standards including screen content coding, light-field video coding, and 360-degree omnidirectional video coding.

Dr. Chan serves as an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING. He was the Secretary of the 2010 IEEE International Conference on Image Processing. He was also the Special Sessions Co-Chair and the Publicity Co-Chair of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, and the Technical Program Co-Chair of the 2014 International Conference on Digital Signal Processing.

Wei Kuang (S'18) received the B. S. degree in School of Electronic and Optical Engineering from Nanjing University of Science and Technology, Nanjing, China, in 2015. Now, he is currently pursuing the Ph.D. Degree in electronic and information engineering at The Hong Kong Polytechnic University. His research interests include machine learning and computer vision in video coding**.**

Wan-Chi Siu (S'77-M'77-SM'90-F'12-Life-F'16) received the MPhil and PhD degrees from The Chinese University of Hong Kong in 1977 and Imperial College London in 1984. He is Life-Fellow of IEEE, Fellow of IET and HKIE, and President (2017-2018) of APSIPA (Asia-Pacific Signal and Information Processing Association). Prof. Siu is now Emeritus Professor, and was Chair Professor, Director of Signal Processing Research Centre, Head of Electronic and Information Engineering Department and Dean of Engineering Faculty of The Hong Kong Polytechnic University. He is an expert in DSP, fast algorithms, super-resolution imaging, 2D and 3D video coding, and machine learning for object recognition and tracking. He has published 500 research papers (over 200 are international journal papers) and has 9 recent patents granted. Prof. Siu was also an independent non-executive director (2000-2015) of a publicly-listed video surveillance company and convenor of the First Engineering/IT Panel of the RAE(1992/93) in Hong Kong. He is an outstanding scholar, with many awards, including the Best Faculty Researcher Award and IEEE Third Millennium Medal (2000). Prof. Siu is/was Guest Editor/Subject Editor/AE for IEEE Transactions on CAS, IP, CSVT, and Electronics Letters, and organized very successfully over 20 international conferences including IEEE society-sponsored flagship conferences, such as TPC Chair of ISCAS1997 and General Chair of ICASSP2003 and General Chair of ICIP2010. He was Vice-President, Chair of Conference Board and Core Member of Board of Governors (2012-2014) of the IEEE Signal Processing Society, and is now a member of IEEE Fourier Award for Signal Processing Committee and some other IEEE committees.