SIP (2015), vol. 4, e10, page 1 of 12 © The Authors, 2015.

doi:10.1017/ATSIP.2015.11

INDUSTRIAL TECHNOLOGY ADVANCES

Overview of HEVC extensions on screen content coding

SHAN LIU[1], XIAOZHONG XU[1], SHAWMIN LEI[1,2] AND KEVIN JOU[2]

Technologies for coding non-camera-captured video contents have received great interests lately due to the rapid growth of application areas such as wireless display and screen sharing, etc. In response to the market demands, the ITU-T Video Coding Expert Group and ISO/IEC Motion Picture Expert Group have jointly launched a new standardization project, i.e. the High-Efficiency Video Coding (HEVC) extensions on screen content coding (HEVC SCC). Several new video coding tools, including intra block copy, palette coding, adaptive color transform, and adaptive motion resolution, have been developed and adopted into HEVC SCC draft standard. This paper reviews the main features and coding technologies in the current HEVC SCC draft standard, with discussions about the performance and complexity aspects compared with prior arts.

Keywords: Video coding standard; HEVC; Screen content coding

Received 30 March 2015; Revised 8 August 2015

I. INTRODUCTION

According to Cisco Visual Networking Index [1], "The sum of all forms of video (TV, video on demand, Internet, and P2P) will be in the range of 80–90% of global consumer traffic by 2018". Driven by the increasing demands, new video coding technologies have been developed, aiming at providing real-time, low-delay, and high-quality video "anywhere and anytime". The High-Efficiency Video Coding (HEVC) [2] is the most recent international video coding standard jointly developed by ITU-T Video Coding Expert Group (VCEG) and ISO/IEC Motion Picture Expert Group (MPEG). Compared with H.264/MPEG-4 AVC standard [3], the preceding project by the same standardization organizations, HEVC can achieve equivalent subjective video quality with around 50% bit-rate reduction [4]. Beyond the completion of its first version in early 2013, which contains Main, Main 10, and Main Still Picture profiles and mainly targets at dealing with 4:2:0 contents, several extensions of HEVC have been developed. In the late 2014, HEVC version 2 was approved by ITU-T which added 21 range extensions (RExt) profiles, two scalable extensions profiles, and one multi-view extensions profile.

All profiles and coding tools in HEVC versions 1 and 2, as well as predecessor standards such as H.264/MPEG-4 AVC and MPEG-2 were developed with a main focus on dealing with camera-captured contents. They may not be as efficient when the source signal characteristics of a specific application deviate from the assumptions that are valid for camera-captured natural videos. Screen content, or non-camera-captured video, is such type of content which usually features computer-generated text and graphics, and sometimes a mixture of computer-generated text, graphics, and camera-captured video. With their distinct signal characteristics, the varied level of the human's visual sensitivity to distortion in different types of content needs to be re-evaluated; visually or mathematically lossless quality may be required. These are some of the challenges imposed to conventional video-coding solutions.

On the other hand, with continuous advancements made in semiconductors, networking, communications, displays, computers, and devices such as tablets and smart phones, real-time, low-delay transport of screen content video between devices are becoming prevalent in many applications, such as screen sharing, wireless display, mobile or external display interfacing, and cloud gaming. These new applications create an urgent need for efficient coding of screen content video. The need is even stronger when these emerging applications become a mainstream from once a niche.

Recognizing the demand and necessity for an industry standard for coding screen content video, ISO/IEC MPEG has released a requirement document in January 2014 for future screen content coding technologies [5], in which

[1]MediaTek USA Inc., San Jose, CA 95134, USA
[2]MediaTek Inc., No.1, Dusing 1st Road, Hsinchu Science Park, Hsinchu City 30078, Taiwan

Corresponding author:
S. Liu
Email: shan.liu@mediatek.com

several types of visual data are required to be supported, including text and graphics with motion, mixed content, animation, and the natural content. The codec design is expected to be "based on the framework of HEVC Main 10 4:4:4, with low extra complexity". The video quality is expected to be "up to visually lossless or mathematically lossless". At the same time, a "Joint Call for Proposals for Coding of Screen Content" has been issued by VCEG and MPEG together [6], which officially launched the HEVC Extensions on Screen Content Coding (HEVC SCC for short) standardization process.

Seven responses to the "Joint Call for Proposals for Coding of Screen Content" were submitted to the 14th JCT-VC Meeting held in Valencia, Spain, in April 2014. The proposed techniques for coding screen content video have been extensively studied and evaluated during the last several meeting cycles. Some coding tools have been included in HEVC SCC working draft and will be included in the Proposed Draft Amendment which is being developed at this moment. The standard is expected to be finalized in February 2016.

This paper provides an overview of the ongoing HEVC SCC draft standard. It is organized as follows. Section II introduces some key technologies in the current HEVC SCC design. Section III presents the performance of current HEVC SCC in comparison with H.264/MPEG-4 AVC High 4:4:4 Predictive profile and HEVC RExt Main 4:4:4 profile, as well as the performance of each individual tool in the current HEVC SCC design. Section IV discusses the underlining complexity considerations for implementing HEVC SCC. Section V gives a list of subjects that is currently under investigation. The paper is concluded in Section VI.

II. OVERVIEW OF HEVC SCC TECHNOLOGIES

As the extensions to HEVC standard with a primary target on applications with non-camera-captured video content, HEVC SCC inherited the coding tools from HEVC version 1 as well as HEVC RExt as the basis for its development. In addition, the characteristics of screen content video have been carefully studied and the following screen content-specific coding tools have been proposed and included in the current working draft of HEVC SCC [7]. In this section, they are briefly introduced primarily from a standard decoder point of view. More information of the encoder algorithm designs for these tools in the reference software may be found in [8].

A) Intra block copy (intraBC)

Motion compensation is one of the key technologies in modern video coding. The correlation between adjacent pictures has been investigated in various efficient ways in the literature to reduce the bandwidth of representing the video signal. Similar concept has also been tried to allow block matching and copy within the same picture. It was not very successful when applying this concept to camera-captured video. Part of the reasons is that the textual pattern in a spatial neighboring area may be similar to the current coding block but usually with gradual changes over space. It is thus difficult for a block to find an exact match within the same picture; therefore the improvement in coding performance is limited. However, the spatial correlation among pixels within the same picture is different for screen content. For a typical video with text and graphics, there are usually repetitive patterns within the same picture. Hence, intra-picture block copy becomes possible and has been proved to be very effective. A new prediction mode, i.e. intraBC mode, is thus introduced to utilize this characteristic. In the intraBC mode, a prediction unit (PU) is predicted from a previously reconstructed block within the same picture. Similar to a PU in motion compensation, a displacement vector (called block vector or BV) is used to signal the relative displacement from the position of the current PU to that of the reference block. The prediction errors after compensation are then coded in the same fashion as how the inter residues are coded in HEVC version 1. An example of intraBC compensation is illustrated in Fig. 1.

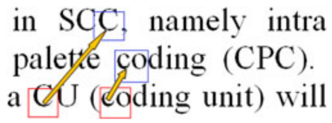


Fig. 1. An example of intra block copy and compensation.

Despite the similarity, there are a few aspects that differentiate the intraBC mode from the inter mode in HEVC version 1:

1) Block partitioning

When first proposed in [9], intraBC can only be applied to PU size of 2N × 2N, i.e. the whole coding unit (CU). It was realized later that when intraBC is applied to sub-CU partitions, significantly higher coding gain can be achieved. Therefore in the current SCC design, intraBC can also be applied to other PU sizes, e.g. 2N × 2N, N × 2N, and N × N, besides 2N × 2N [10]. Note that N × N intraBC is only allowed when the current CU is a smallest CU.

2) BV prediction and coding

New methods for predicting and coding intraBC BVs are proposed, as opposed to the methods for predicting and coding motion vectors in HEVC version 1, for further improving the coding efficiency of HEVC SCC. In the new BV prediction scheme, one candidate from the left and one from the top neighbors are used as the two primary predictors for BV prediction [11, 12]. In the case when spatial neighbors are not available, e.g. neighboring blocks are coded in regular intra or inter mode instead of the intraBC mode, or hit the picture or slice boundaries, two last coded BVs are used to fill the candidate list. At the beginning of each coding tree unit (CTU), these two so-called last coded BVs are initialized using constant values. A 1-bit flag is used to signal one of the two candidates that is used to predict

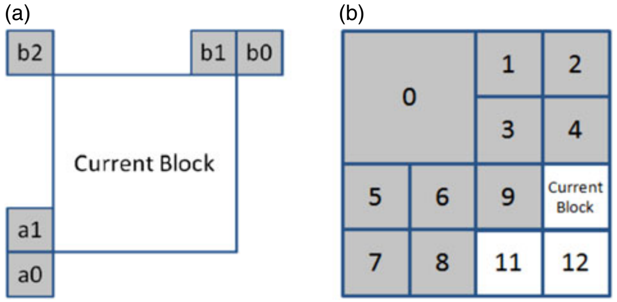


Fig. 2. An example of intra block copy, block vector prediction. (a) Spatial neighboring positions, where a1 and b1 positions are selected as the spatial candidates for block vector prediction. (b) An example of BV predictor candidate list construction.

the current BV. As illustrated in Fig. 2(a), the two spatial neighboring positions a1 and b1 are used as the primary predictor candidates. Figure 2(b) shows an example of intraBC BV predictor list construction, where the spatial neighbors for the current block are block 9 and block 4. If only blocks 2 and 8 are intraBC coded while all others are not, then the BVs of these two blocks are considered as two last coded BVs and used as the predictor candidates for the current block. If in addition block 4 is intraBC coded, then the BVs of block 4 (spatial neighbor) and block 8 (last coded) are used as the two BV predictor candidates.

The BV prediction difference needs to be further entropy coded. A new binarization method was proposed in [13], as opposed to the binarization method for coding the motion vector prediction difference as in HEVC version 1. In this method, a 1-bit flag is context coded to signal whether the BV component being coded, either x or y, is 0. If it is not zero, the magnitude of this component is bypass coded using Exponential-Golomb code order 3, followed by its sign.

3) Search range consideration

Intra picture block copy is a causal process, that is, an intraBC block can only use the previously reconstructed pixels in the same picture as its predictor. When intraBC was first proposed, the search range was kept local. That is, only pixels from the CTU that the current block belongs to and its left neighboring CTU can be used as intraBC predictor. Later on, it was decided to extend the search area to the full picture [14]. That is, all previously reconstructed pixels in the current picture can be used as predictor. In order for a BV to be valid, at least one of the following conditions shall be true:

$$\mathrm{BV\_}x + n\mathrm{PbSw} + x\mathrm{PbS} - x\mathrm{Cbs} <= 0, \qquad (1)$$

$$\mathrm{BV\_}y + n\mathrm{PbSh} + y\mathrm{PbS} - y\mathrm{Cbs} <= 0, \qquad (2)$$

where (BV_x, BV_y) is the BV for the current PU; nPbSw and nPbSh are the width and height of the current PU; (xPbS, yPbS) is the location of the top-left pixel of the current PU relative to the current picture; and (xCbs, yCbs) is the location of the top-left pixel of the current CU relative to the current picture.

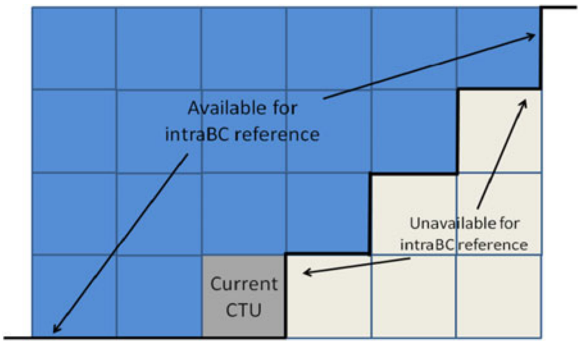

Available for intraBC reference
Unavailable for intraBC reference
Current CTU

Fig. 3. Demonstration of the search range constraint in (3) for intra block copy.

Furthermore, when wavefront parallel processing (WPP) is enabled, some regions top-right to the current block cannot be used for prediction as the pixels in these regions may not yet be decoded and reconstructed while the current block is being processed. Therefore, one additional constraint is applied to regulate block vectors as proposed in [15, 16], and described as follows:

$$\begin{aligned}&(x\mathrm{Pb} + \mathrm{BV\_}x + n\mathrm{PbSw} - 1)/\mathrm{CtbSizeY}\\&\quad - x\mathrm{Cbs}/\mathrm{CtbSizeY} <= y\mathrm{Cbs}/\mathrm{CtbSizeY}\\&\quad - (y\mathrm{Pb} + \mathrm{BV\_}y + n\mathrm{PbSh} - 1)/\mathrm{CtbSizeY}, \qquad (3)\end{aligned}$$

where CtbSizeY is the size of the CTU. In fact, constraint (3) only causes very small coding performance decrease when WPP is not used, thus it is decided that constraint (3) is always enforced regardless of whether WPP is enabled. Figure 3 demonstrates when constraint (3) is invoked, in which the shaded CTUs top-left to the current CTU can be used for intraBC prediction of any block belongs to the current CTU, while the rest CTUs which are bottom-right to the current CTU are unavailable for the same prediction.

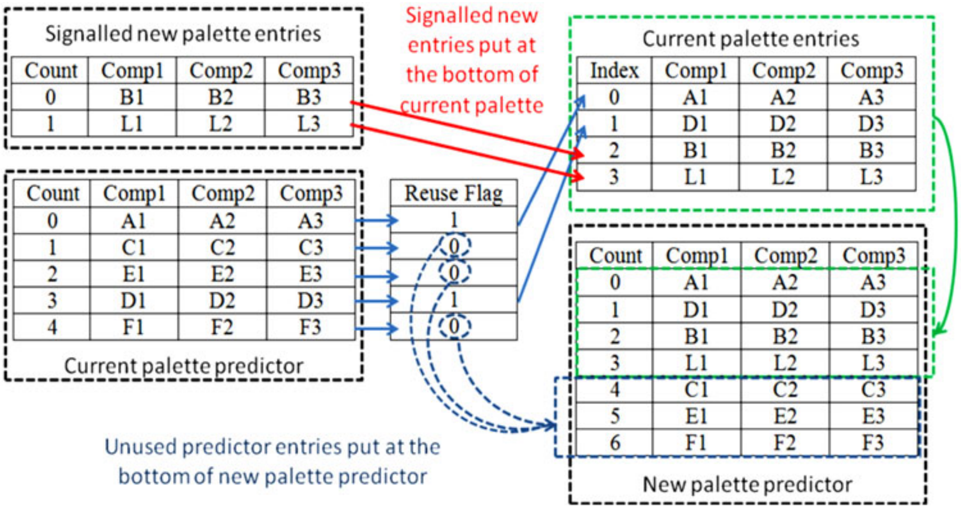
It is also asserted that intra picture prediction should not cross-slice or tile boundaries. Therefore, a valid BV shall not point to areas in other slices or tiles than the current one.

4) Filtering operations

In current SCC design, intra picture block copy and prediction are constrained to use integer BV resolution, thus no interpolation filter is applied. Furthermore, when the reconstructed pixels in the current picture are used as intraBC predictor, they bypass in-loop filters such as sample adaptive offset filter and deblocking filter. The filtered pixels are used for temporal (motion) predictor for later coded picture and final output. Hence both versions of reconstructed current picture, filtered and unfiltered, need to be temporarily stored. After the coding of the current picture is completed, the storage of the unfiltered version may be released.

B) Palette coding

Besides the repetition of textual patterns inside one picture, another unique characteristic of screen content is the statistically fewer number of colors used for representing an image block, in comparison with an image block of the same


Signalled new palette entries
Signalled new entries put at the bottom of current palette
Current palette entries
Current palette predictor
Unused predictor entries put at the bottom of new palette predictor
New palette predictor

Count	Comp1	Comp2	Comp3
0	B1	B2	B3
1	L1	L2	L3

Index	Comp1	Comp2	Comp3
0	A1	A2	A3
1	D1	D2	D3
2	B1	B2	B3
3	L1	L2	L3

Count	Comp1	Comp2	Comp3
0	A1	A2	A3
1	C1	C2	C3
2	E1	E2	E3
3	D1	D2	D3
4	F1	F2	F3

Reuse Flag
1
0
0
1
0

Count	Comp1	Comp2	Comp3
0	A1	A2	A3
1	D1	D2	D3
2	B1	B2	B3
3	L1	L2	L3
4	C1	C2	C3
5	E1	E2	E3
6	F1	F2	F3

Fig. 4. Use of palette predictor to signal palette entries.

size in camera-captured content. For example, in a screen content image with text, typically a coding block contains only the foreground text color and the background color. Sometimes, the random patterns of text characters and letters make it challenging for the current coding block to find a matching block in the same or previously coded pictures. It may also be challenging to utilize the directional local intra prediction for efficient compression in this circumstance. A new coding tool, a.k.a. palette coding was proposed and proven to be effective for handling this type of source.

Briefly speaking, palette coding is a major color-based prediction method [17, 18]. All pixels (their values of all components, Y/Cb/Cr or R/G/B) of a coding block are classified into a list of major colors. A major color is a representative color which has high frequency of occurrence in the block. For each palette coded CU, a color index table, i.e. palette is formed with each index entry associated with three (Y/Cb/Cr or R/G/B) sample values. All the pixels in the CU are converted into corresponding indices, except some rarely used pixels, which are isolated in the color index map and cannot be quantized to any of the major colors. These pixels are called escape pixels and an ESCAPE symbol is used to mark each of them. The actual pixel values of these escape pixels are signaled explicitly. The indices, including ESCAPE, are run-length coded using the index of either above or left neighbor as predictor. The following paragraphs describe how palette coding is performed with more detail.

1) Coding of palette entries

The coding of palette entries for a CU is based on a predictor buffer called palette predictor. For each entry in the palette predictor, a reuse flag is sent to signal whether or not this entry will be used in the current palette. If yes, this entry will be put in front of the current palette. For those entries in the current palette but not in the palette predictor, the number of them and their pixel values are signaled. These signaled new entries are put at the bottom of the current palette. The current palette size is then calculated as the number of reused palette entries plus the number of signaled palette entries.

After decoding a palette-coded CU, the palette predictor is updated for the next palette-coded CU. This is done using the information of the current palette. The entries of the current palette are put in front of the new palette predictor, followed by those unused entries from the previous palette predictor. This process is called "palette predictor stuffing". The new predictor size is then calculated as the size of the current palette plus the number of unused palette entries. In Fig. 4, an example of the current palette derivation and the palette predictor update is shown. In this example, the palette predictor for the current CU has a size of 4, two of which are reused. The current palette has a size of 4, while the updated predictor (for the next CU) has a size of 7.

For a special case, when the current palette is exactly the same as the one previously coded (the last coded palette), a sharing flag is used to indicate this scenario. In this case, no new palette entry signaling or palette predictor update is necessary.

Note that the maximum sizes for the palette and palette predictor are signaled at the SPS (sequence parameter set) level. In the reference software, they are set to be 31 (plus one for escape index) and 64, separately. The update of palette predictor will stop if the maximum predictor size is reached. The last coded palette and palette predictor is set to zero size as an initialization at the beginning of each slice or, they are treated in a similar way as the CABAC status synchronization at the beginning of each WPP thread [15, 19]. That is, at the beginning of each CTU row, the last coded palette and palette predictor information for the first palette-coded CU will be set to the last used palette and palette predictor from the top-right CTU relative to the current CTU.

2) Coding of palette indices

The palette indices are coded sequentially using either horizontal or vertical traverse scan as shown in Fig. 5. The scan order is explicitly signaled. As the vertical scan can be regarded as horizontal scan over a transposed CU, it is

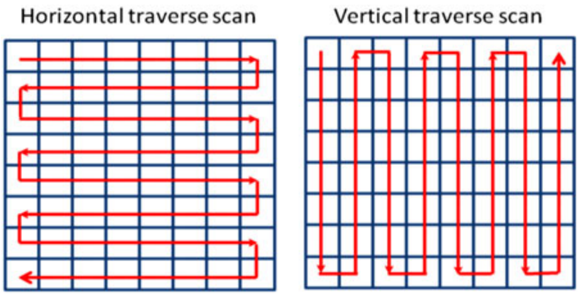


Fig. 5. Horizontal and vertical traverse scans.

assumed that the horizontal scan is used in this paper for simplicity.

The palette indices are coded using two prediction modes: “COPY_INDEX” and “COPY_ABOVE”. The ESCAPE symbol is treated as the largest index. In the “COPY_INDEX” mode, the palette index is explicitly signaled. In the “COPY_ABOVE” mode, the palette index of the sample in the row above is copied. For both “COPY_INDEX” and “COPY_ABOVE” modes, a run value is signaled which specifies the number of subsequent samples that are also coded using the same mode. For an ESCAPE symbol, its pixel value (all components) needs to be signaled. The coding of palette indices is illustrated in Fig. 6.

3) Other aspects of palette coding

There is no residue coding for palette coding mode. The pixel reconstruction is just to convert the decoded palette indices into pixels according to the current palette. Further, no deblocking filter is used at a palette-coded CU boundary.

C) Adaptive color transform (ACT)

An ACT operates at the CU level, where a 1-bit flag is used to signal whether a color space transform is used for each of the prediction residue pixels in this CU. The motivation of this tool is that for a given color space (e.g. RGB space), there are certain correlation among different components of the same pixel. Even after the prediction from a spatial or temporal neighboring pixel, the correlation among the residue pixel components still exists. A transform of color space may be helpful to concentrate the energy and therefore improve the coding performance. Note that in palette coding, ACT does not apply as there is no residue coding. A brief overview of ACT for lossy [20] and lossless [21] coding will be given in this subsection.

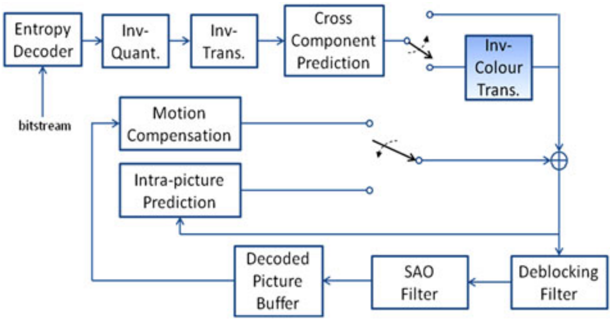


Fig. 7. High Efficiency Video Coding extensions on screen content coding decoder flow with adaptive color transform.

1) ACT in lossy coding

For the prediction residue of each pixel, a color space transform is performed, prior to the compensation from intra or inter prediction. A decoder flow of ACT is shown in Fig. 7. In lossy coding, ACT uses the followin color space conversion equations:

$$\text{Forward:}\ \begin{bmatrix} C_0' \\ C_1' \\ C_2' \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} \Big/ 4$$

$$\text{Inverse:}\ \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} C_0' \\ C_1' \\ C_2' \end{bmatrix}, \qquad (4)$$

where (C_0, C_1, C_2) and (C_0', C_1', C_2') are the three color components before and after color space conversion, respectively. The forward color transform from (C_0, C_1, C_2) to (C_0', C_1', C_2') is not normalized. In order to compensate for the non-normalized nature of the forward transform, for a given normal QP value for the CU, if ACT is turned on, the quantization parameter is set equal to (QP – 5, QP – 3, QP – 5) for (C_0', C_1', C_2'), respectively. The adjusted quantization parameter only affects the quantization and inverse quantization of the residuals in the CU. In the deblocking process, the normal QP value is still used.

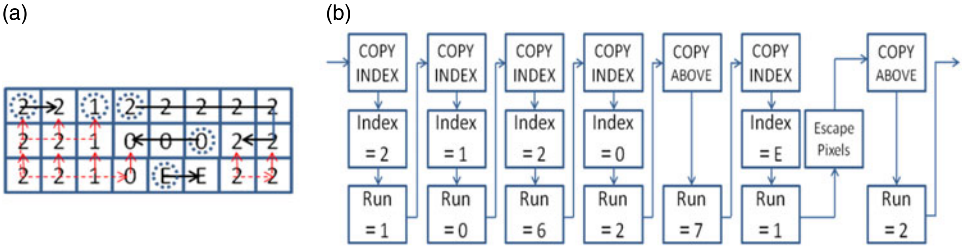


Fig. 6. Example of coding of palette indices. (a) Palette indices and their modes. Dotted circle: signaled indices in “COPY_INDEX” mode; horizontal solid arrow: “COPY_INDEX” mode; vertical solid arrow: “COPY_ABOVE” mode. (b): Decoded string of syntax elements for the indices in (a).

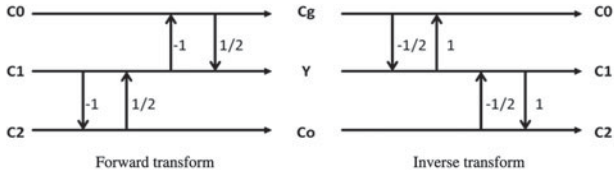


Fig. 8. Flow chart of lifting operations of forward and inverse adaptive color transform in lossless coding.

2) ACT in lossless coding

For lossless coding (where no transform or quantization applies), color space conversion based on the YCoCg-R space (the reversible version of YCoCg) as depicted in Fig. 8 is used in ACT. The reversible color space transform increases the intermediate bit depth by 1-bit after forward transform. Note that the QP adjustment discussed in lossy coding does not apply.

D) Adaptive motion vector resolution

Screen content by definition is not camera-captured. The motion from one screen content image to another should have an integer displacement. Therefore, the aliasing effect due to the camera sampling of a temporal motion may not be valid. If fractional-pel motion compensation is not used at all, then the bits used to present fractional-pel motion vectors can be saved. However, this approach (use integer motion vector resolution always) is apparently not suitable for natural content. In addition, simulation results showed that significant losses will occur even for some screen content sequences, if integer motion resolution is enforced at all times. Therefore, it is proposed in [22] that a slice-level flag is used, to signal that whether the motion vectors in this slice is at integer-pel resolution or quarter-pel resolution. At the decoder side, the decoded motion vector needs to be left shifted by two if "integer motion vector" is used in the slice. As for the encoder design, different approaches were proposed to determine whether integer motion is adequate for the current slice. In the two-pass approach, the current slice is encoded twice using both integer motion and quarter-pel motion; the solution with better RD result is then selected. As a result, the encoder time is doubled. In an alternative approach, a pre-analysis of the slice content is performed using the original pixels. The percentage of 8×8 homogeneous blocks is calculated. The definition of homogeneous block is that it can find the perfect match in the first reference picture of list 0 for the slice, or it has no textual pattern (the whole block is single valued). If the estimated percentage of 8×8 homogeneous blocks is over a pre-defined threshold, the slice-level flag will be set on to use integer motion. By doing this, the two-pass coding process is avoid, while the majority of coding performance gain in two-pass approach can be preserved ($3 \sim 4\%$ BD rate saving) for text and graphics with motion 1080p class test sequences. Note that for camera-captured and animation video, no significant coding benefit can be observed by using this tool.

III. PERFORMANCE ANALYSIS

A) Simulation setup

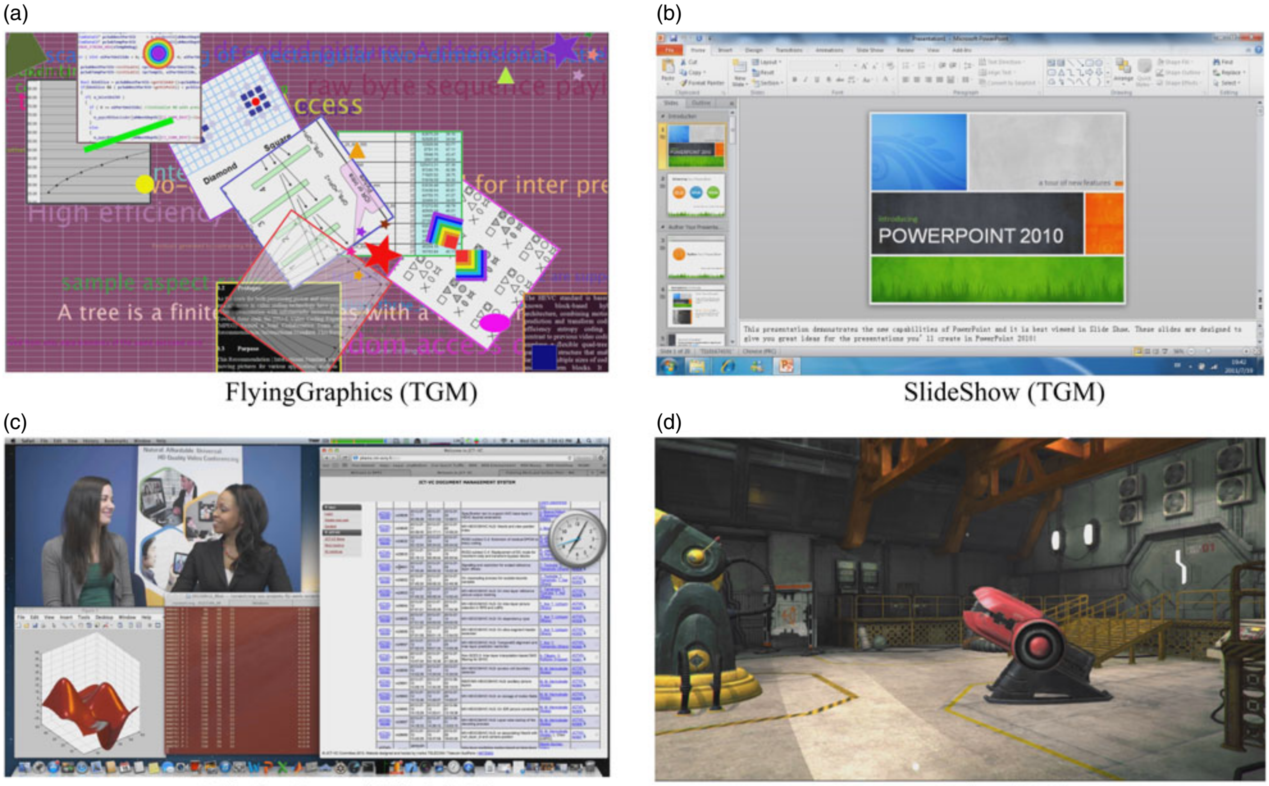
In this section, the coding performance of current HEVC SCC is evaluated using its reference software SCM-3.0 [23], and compared with the coding performance of H.264/MPEG-4 AVC standard High 4:4:4 Predictive profile (reference software JM-18.6 [24]) and HEVC RExt Main 4:4:4 profile (reference software RExt-8.1) [25]. The simulations are conducted under HEVC SCC common test conditions listed in [26], in which a set of non-camera-captured as well as camera-captured video sequences are tested in both RGB 4:4:4 and YUV 4:4:4 color formats. In the text and graphics with motion (TGM) class, seven sequences are selected to represent the most common screen content videos. In the mixed content (MC) class, three sequences are selected containing a mixture of both natural video and text/graphics. One animation (ANI) sequence and two camera-captured (CC) video sequences are also tested. Some screen shots of selected screen content video sequences are shown in Fig. 9. A summary of the test sequences is provided in Table 1. More details about these sequences can be found in [26]. All the sequences are tested under testing configurations of all intra (AI), random access (RA), and low-delay B (LB). The results are measured in B-D rates [27] based on the calculation from four QP points, i.e. QP equals to 22, 27, 32, and 37. Negative values in the tables mean BD rate reductions or coding efficiency improvement. In addition, lossless coding results are also provided. Note that when RGB sequences are coded, the actual color component order is GBR based on the assumption that the human visual system is more sensitive to green color. Also, only the B-D rates of Y/G component of YUV/RGB format is presented in the following tables for illustration.

B) Overall performance evaluation of HEVC SCC

The coding performance of HEVC SCC in comparison with the prior art standards, H.264/MPEG-4 AVC High 4:4:4 Predictive profile [28] and HEVC RExt Main 4:4:4 profile are reported in this section. The relative B-D rate savings of HEVC SCC on top of H.264/MPEG-4 AVC High 4:4:4 Predictive profile and HEVC RExt Main 4:4:4 profile are demonstrated in Tables 2 and 3, respectively.

C) Individual tools evaluation

In addition to the overall performance evaluation, the individual tool evaluation is also performed [28, 29]. Among the four individual tools, the adaptive motion vector resolution is relatively simple and basically there is no technical change since its adoption. Its performance has been reported during the discussion hence no repeated test is performed in this section. For all other coding tools discussed earlier in this paper, including intraBC, palette coding, and ACT are turned off from SCM-3.0 one by one, to show the



FlyingGraphics (TGM)

SlideShow (TGM)

MissionControlClip3 (MC)

Robot (ANI)

Fig. 9. Typical types of screen content video. (a) Flying Graphics (text and graphics with motion (TGM)). (b) Slide Show (TGM). (c) Mission Control Clip 3 (mixed content (MC)). (d) Robot (animation (ANI)).

Table 1. 4:4:4 test sequences.

Resolution	Sequence	Bit depth	Category	Fps
1920 × 1080	FlyingGraphics	8-bit	TGM	60
1920 × 1080	Desktop	8-bit	TGM	60
1920 × 1080	Console	8-bit	TGM	60
1280 × 720	Web_browsing	8-bit	TGM	30
1280 × 720	Map	8-bit	TGM	60
1280 × 720	Programming	8-bit	TGM	60
1280 × 720	SlideShow	8-bit	TGM	20
2560 × 1440	Basketball_Screen	8-bit	MC	60
2560 × 1440	MissionControlClip2	8-bit	MC	60
1920 × 1080	MissionControlClip3	8-bit	MC	60
1280 × 720	Robot	8-bit	ANI	30
1920 × 1080	EBURainFruits	10-bit	CC	50
1920 × 1080	Kimono1	10-bit	CC	24

impacts on the performance. In Tables 4 and 5, the relative B-D rate increments (positive numbers) for turning off each tool are shown for lossy and lossless coding conditions, respectively. Here the anchor is SCM-3.0 with all tools turned on.

D) Discussion

Several remarks can be made from the results presented above. Firstly, HEVC SCC can achieve significantly higher coding efficiency, when compared with the prior art standards, e.g. H.264/MPEG-4 AVC High 4:4:4 Predictive profile and HEVC RExt Main 4:4:4 profile for coding of screen content video. No apparent benefit is observed for animated and camera-captured video contents in YUV format when compared with HEVC RExt Main 4:4:4 profile. Some coding gains are reported for animated and camera-captured video contents in RGB format due to the effect of ACT, while it should be noticed that the coding gains of ACT on B and R color components are reported less than that on G color component [28, 29]. For individual coding tools, intraBC contributes the largest portion of the total coding gain, while palette coding also provides quite substantial improvement when compressing screen content videos. These two tools have limited benefit when dealing with animated and camera-captured contents for either RGB or YUV format. As for ACT, its benefit under different testing configurations vary: for all classes in RGB format for lossy coding, the coding gains provided by ACT are quite substantial; however, for lossy coding in YUV format, as well as for lossless coding in both RGB and YUV formats, the impact of ACT seems very limited.

The above discussion reveals the benefit of HEVC SCC as a potential new standard, as well as the breakdown behavior of each individual coding tool. With this information in mind, an efficient codec may be designed with high performance and least necessary effort, for various video contents and application scenarios.

Table 2. SCM-3.0 performance comparison, lossy.

	H.264/MPEG-4 AVC			HEVC RExt		
Performance relative to testing conditions	AI (%)	RA (%)	LB (%)	AI (%)	RA (%)	LB (%)
RGB, TGM, 1080p & 720p	−85.9	−79.8	−77.2	−64.0	−55.2	−49.6
RGB, mixed content, 1440p & 1080p	−79.9	−74.0	−69.2	−54.4	−49.4	−40.6
RGB, animation, 720p	−52.4	−54.9	−56.4	−26.3	−26.1	−24.5
RGB, camera captured, 1080p	−58.4	−63.2	−60.1	−25.7	−27.9	−26.1
YUV, TGM, 1080p & 720p	−74.3	−67.1	−64.7	−57.0	−45.9	−39.1
YUV, mixed content, 1440p & 1080p	−63.2	−56.2	−50.3	−44.6	−35.3	−21.7
YUV, animation, 720p	−23.4	−32.3	−39.0	−1.2	−0.4	−0.1
YUV, camera captured, 1080p	−26.6	−39.8	−39.8	0.3	0.8	0.6

Table 3. SCM-3.0 performance comparison, lossless.

	H.264/MPEG-4 AVC			HEVC RExt		
Performance relative to testing conditions	AI (%)	RA (%)	LB (%)	AI (%)	RA (%)	LB (%)
RGB, TGM, 1080p & 720p	−69.6	−55.1	−57.3	−48.0	−30.2	−29.2
RGB, mixed content, 1440p & 1080p	−43.8	−28.4	−26.7	−23.5	−6.1	−3.9
RGB, animation, 720p	−20.9	−14.8	−13.2	−4.5	−1.3	−1.3
RGB, camera captured, 1080p	−9.2	−3.7	−3.5	−0.4	0.3	0.3
YUV, TGM, 1080p & 720p	−59.7	−44.9	−46.4	−50.6	−31.1	−29.1
YUV, mixed content, 1440p & 1080p	−28.9	−16.6	−14.9	−23.5	−6.0	−3.7
YUV, animation, 720p	−5.1	−8.1	−6.3	−1.9	−0.3	−0.3
YUV, camera captured, 1080p	−0.9	−1.6	−1.5	0.0	0.0	0.0

Table 4. SCM-3.0 with single tool off versus SCM-3.0, lossy.

	intraBC off			Palette off			ACT off		
Tools turned off testing conditions	AI (%)	RA (%)	LB (%)	AI (%)	RA (%)	LB (%)	AI (%)	RA (%)	LB (%)
RGB, TGM, 1080p & 720p	55.6	28.4	13.0	19.4	12.4	7.5	10.9	13.6	13.1
RGB, mixed content, 1440p & 1080p	41.1	23.9	9.9	3.9	2.6	1.5	20.0	30.3	31.6
RGB, animation, 720p	1.1	0.5	0.1	0.0	0.1	0.0	32.4	33.1	31.6
RGB, camera captured, 720p	0.1	0.0	0.0	0.0	0.0	0.0	32.8	38.2	35.7
YUV, TGM, 1080p & 1080p	59.7	29.2	11.7	20.3	13.0	7.5	0.4	0.7	1.1
YUV, mixed content, 1440p & 1080p	42.5	25.0	10.0	6.3	4.6	2.8	−0.1	−0.4	−0.4
YUV, animation, 720p	1.6	0.5	−0.1	−0.1	−0.2	−0.1	−0.1	0.1	0.0
YUV, camera captured, 1080p	0.2	0.0	−0.1	0.0	−0.1	−0.1	−0.1	−0.5	−0.3

Table 5. SCM-3.0 with single tool off versus SCM-3.0, lossless.

	intraBC off			Palette off			ACT off		
Tools turned off testing conditions	AI (%)	RA (%)	LB (%)	AI (%)	RA (%)	LB (%)	AI (%)	RA (%)	LB (%)
RGB, TGM, 1080p & 720p	27.1	24.6	25.8	14.2	3.0	2.5	0.1	−0.2	−0.2
RGB, mixed content, 1440p & 1080p	20.4	3.8	2.2	1.2	0.2	0.0	1.8	0.5	0.5
RGB, animation, 720p	0.1	0.0	0.0	0.0	0.0	0.0	4.4	1.3	1.3
RGB, camera captured, 1080p	0.0	0.0	0.0	0.0	0.0	0.0	0.4	−0.3	−0.3
YUV, TGM, 1080p & 720p	30.1	25.9	26.0	13.9	2.8	2.1	0.1	0.3	0.3
YUV, mixed content, 1440p & 1080p	23.5	4.4	2.5	1.0	0.1	0.0	0.0	0.0	0.0
YUV, animation, 720p	0.1	0.0	0.0	0.0	0.0	0.0	1.6	0.3	0.3
YUV, camera captured, 1080p	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

IV. COMPLEXITY ANALYSIS

A) IntraBC

In motion compensation, the reference picture data are typically stored off-chip and put on-chip from time to time when necessary. In the hardware design, the off-chip memory is usually organized in group, e.g. in blocks of 8 × 2, etc. When the motion vector points to a particular position, all the memory blocks that contain at least one pixel of the reference block need to be accessed and loaded on chip. When interpolation is taken into consideration, an extra few lines surrounding the reference block also need to be accessed. Therefore, when a compensation block size is small, the overhead of reading unused pixels from memory blocks in the worst case is high. The memory bandwidth consumption is measured in (5) [30]:

$$P = \frac{\lceil (m-1+M+L-1)/m \rceil \cdot \lceil (n-1+N+L-1)/n \rceil \cdot m \cdot n}{M \cdot N}, \tag{5}$$

where M and N respectively represent the width and height of the smallest unit for intra copying, m and n denote respectively the width and height of the memory access pattern (e.g. 4×2, 8×4, etc.), and L is related to the tap length of interpolation filter (e.g. $L = 8$ for an 8-tap filter and $L = 1$ for no interpolation). Compared with the existing HEVC version 1, there is no interpolation needed for an intraBC coded PU. In light of this fact, even though the 4×4 intraBC block is smaller than the smallest partitions in HEVC ($8 \times 4/4 \times 8$ partitions), the worst case in memory bandwidth consumption for the whole codec design is still maintained.

Another aspect of the memory bandwidth issue for intraBC is that there is some additional memory bandwidth requirement for extra reading/writing operation for unfiltered reference picture. Because the full-frame search is enabled in intraBC, previously reconstructed, but unfiltered samples need to be written off-chip for storage and read on-chip for compensation. Compared with a regular reference picture for inter motion compensation, this part is the extra effort.

Other aspects of the intraBC mode complexity are similar to the inter mode in HEVC.

B) Palette coding

As mentioned in the introduction of palette coding, no residue coding (transform or quantization) is performed for palette coding. In this aspect, the complexity of palette decoding is reduced, as compared with intra mode or inter mode in HEVC. However, from the parallel processing point of view, the operation of palette coding is per pixel basis, that is, in the worst case each pixel in a CU needs to be processed separately (assigning index, parsing run length, and converting index to pixel value). This is different from the line-based processing in intra mode and block-based processing in inter mode.

C) ACT

The extra operation this tool introduces is a one-dimensional transform per pixel (among three components). As for the encoder design, it adds one more dimension of mode decision loop to decide whether this tool should be used or not. In a brute-force decision approach, the computations will be doubled in mode decision. According to the results shown in Section IV and some previous study [31], it would be a practical solution to link this CU-level decision to the color format of the video. In particular, one can always enable this tool for all CUs in RGB format video and always disable it for YUV format.

D) Adaptive motion vector resolution

The extra complexity of this tool over the existing HEVC inter mode is negligible. For the encoder design, in order to avoid 2-pass coding, the pre-analysis of the slice is necessary, which requires some data access of the whole slice before the block-by-block processing of it.

V. FURTHTER DEVELOPMENT

A) Unification of intraBC mode and inter mode

It was proposed to consider the (partially) reconstructed current picture as an additional reference picture as opposed to conventional temporal reference pictures, such that the intraBC mode signaling can be unified with inter mode [32–34]. Consequently, the indication of the use of intraBC mode is via a reference index in the reference picture list instead of an intraBC mode flag in each CU. Then when a PU's reference picture index points to this particular reference picture, this PU is intraBC coded. This approach has been studied for a few meeting cycles and was included in the draft standard in February 2015 [35]. With this adoption, the current picture is regarded as a reference picture (for intraBC mode) and is put in the last position of reference list 0 by default. In addition, merge, skip, and AMVP modes may be enabled when the reference picture is current picture. When unified with inter mode, no very significant coding performance change is observed compared with intraBC as a CU mode [35]. One main benefit identified for unifying intraBC with inter mode is that a large part of existing HEVC version 1 inter mode design can be shared and used for intraBC. In some implementations intraBC can be enabled with a few high-level changes and bitstream conformance constraints (e.g. (1) – (3) in Section II.A.3).

B) Extended to non-4:4:4 formats

HEVC SCC was initially designed for 4:4:4 color format only. It is under development to further extend the scope to cover potential non-4:4:4 applications (including monochrome format) as well. For this purpose, the palette coding has provided necessary supports for coding of non-4:4:4 video formats [36, 37] [38], while the intraBC mode has already covered non-4:4:4 formats in its current design. The ACT will not be used in non-4:4:4 coding scenario.

C) Other tools experimented

One extension of intraBC is intra line copy (ILC) [39], in which an intraBC PU can be further split into multiple lines and the prediction is performed on each line. The length of each line may be either 2N or N, depending on the direction of the line and the size of the PU. For example, a 2N × N PU

Table 6. Memory bandwidth assessment for different methods.

Memory pattern ($m \times n$)	Sum of memory bandwidth (three components)				
	4 × 2	4 × 4	8 × 2	8 × 4	8 × 8
8 × 8 HEVC Bi-pred	24	28.5	33	39	54
4 × 4 intraBC	9	12	18	24	48
1 × 4 ILC	18	24	36	48	96
1 × 1 ISC	24	48	48	96	192

may be split into either N (2N × 1) horizontal lines or 2N (N × 1) vertical lines. Each line has a line vector, pointing to the location of its predictor line within the current picture.

Similar to the run-length coding in palette coding, intra string copy (ISC) [40] is proposed to HEVC SCC, in which a variable length of connected pixels (string) in scan order is predicted from reconstructed pixels with same pattern. The offset from the first pixel of this string to the first pixel of the predictor string is signaled, followed by the length of the string.

On top of SCM-3.0, additional coding performance gains are observed from using ILC or ISC. One issue currently under investigation is the memory bandwidth of these methods in the worst-case scenario. The worst-case memory bandwidth consumptions of ILC and ISC are presented in Table 6, compared with the worst-case memory bandwidth consumptions of HEVC version 1 and intraBC [41] calculated by (5). It is desired to reduce the worst-case memory bandwidth consumption of these methods such that they can be part of a practical design. As the standardization approaches its finalization stage, these experimented new tools are less likely to be included in the standard.

VI. SUMMARY

This paper provides an overview of the ongoing draft standard HEVC extensions on SCC. Several key aspects of it are introduced and discussed, including motivation, timeline, algorithm design, performance, and complexity analysis. From the reported results, the current HEVC SCC significantly outperforms the prior art standards, e.g. H.264/MPEG-4 AVC High 4:4:4 Predictive profile and HEVC RExt Main 4:4:4 profile, in terms of coding efficiency. The additional design complexity on top of HEVC version 1 has been carefully examined and kept minimized. In overall, the ongoing HEVC SCC provides a good coding efficiency and complexity trade-off for coding screen content videos.

REFERENCES

[1] Cisco Visual Networking Index: Forecast and Methodology: 2013–2018, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html

[2] Sullivan, G.; Ohm, J.; Han, W.-J.; Wiegand, T.: Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Trans. Circuits Syst. Video Technol., **22** (12) (2012), 1649–1668.

[3] Wiegand, T.; Sullivan, G.J.; Bjøntegaard, G.; Luthra, A.: Overview of the H.264/AVC video coding standard. Circuits Syst. Video Technol., **13** (7) (2003), 560–576.

[4] Ohm, J.; Sullivan, G.; Schwarz, H.; Tan, T.; Wiegand, T.: Comparison of the coding efficiency of video coding standards – including High Efficiency Video Coding (HEVC). IEEE Trans. Circuits Syst. Video Technol., **22** (12) (2012), 1669–1684.

[5] Yu, H.; McCann, K.; Cohen, R.; Amon, P.: Requirements for Future Extensions of HEVC in Coding Screen Content. ISO/IEC JTC1/SC29/WG11 MPEG2014/N14174, January 2014.

[6] ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio: Joint Call for Proposals for Coding of Screen Content. ISO/IEC JTC1/SC29/WG11 MPEG2014/N14175, January 2014.

[7] Joshi, R.; Xu, J.: HEVC Screen Content Coding Draft Text 2. JCTVC-S1005, October 2015.

[8] Joshi, R.; Xu, J.; Cohen, R.; Liu, S.; Ma, Z.; Ye, Y.: Screen Content Coding Test Model 3 Encoder Description (SCM 3). JCTVC-S1014, October 2014.

[9] Budagavi, M.; Kwon, D.-K.: AHG8: Video Coding using INTRA Motion Compensation. JCTVC-M0350, April 2013.

[10] Chang, T.-S. *et al.*: RCE3: Results of Subtest D.2 on Nx2N/2NxN/NxN Intra Block Copy. JCTVC-P0180, January 2014.

[11] Pang, C. *et al.*: Non-SCCE1: Combination of JCTVC-R0185 and JCTVC-R0203. JCTVC-R0309, July 2014.

[12] Xu, X.; Liu, S.; Chuang, T.-D.; Lei, S.: Block Vector Prediction in Intra Block Copy for HEVC Screen Content Coding, in accepted by Data Compression Conf. 2015.

[13] Pang, C.; Rapaka, K.; Sole, J.; Karczewicz, M.: SCCE1: Test 3.6 – Block Vector Coding for Intra Block Copy. JCTVC-R0186, July 2014.

[14] Pang, C.; Sole, J.; Hsieh, T.; Karczewicz, M.: Intra Block Copy with Larger Search Region. JCTVC-Q0139, April 2014.

[15] Li, B.; Xu, J.: On WPP with Palette Mode and Intra BC Mode. JCTVC-S0088, October 2014.

[16] Lai, P.; Xu, X.; Liu, S.; Chuang, T.-D.; Lei, S.: AHG14: Intra Block Copy Reference Area for Wavefront Parallel Processing (WPP). JCTVC-S0101, October 2014.

[17] Guo, L.; Pu, W.; Zou, F.; Sole, J.; Karczewicz, M.; Joshi, R.: Color Palette for Screen Content Coding, in 2014 IEEE Int. Conf. on Image Processing (ICIP), October 2014, 5556–5560.

[18] Onno, P.; Xiu, X.; Huang, Y.-W.; Joshi, R.: Suggested Combined Software and Text for Run-Based Palette Mode. JCTVC-R0348, July 2014.

[19] Misra, K.; Kim, S.H.; Segall, A.: Using the Wavefront Store-and-Sync Design *for Palette Table Prediction Variables*. JCTVC-S0141, October 2014.

[20] Zhang, L. *et al.*: SCCE5 Test 3.2.1: In-Loop Color-Space Transform. JCTVC-R0147, July 2014.

[21] Henrique Malvar, S.; Sullivan, G.J.; Srinivasan, S.: Lifting-based reversible color transformations for image compression. SPIE Proc., **7073** (2008), DOI:10.1117/12.797091.

[22] Li, B.; Xu, J.; Sullivan, G.; Zhou, Y.; Lin, B.: Adaptive Motion Vector Resolution for Screen Content. JCTVC-S0085, October 2014.

[23] HEVC SCC Reference Software Version 3.0 (SCM-3.0): [Online]. Available: https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-16.2%2BSCM-3.0

[24] H.264/AVC Reference Software Version 18.6 (JM-18.6): http://iphome.hhi.de/suehring/tml/download/

[25] HEVC SCC Reference Software Version 8.1 (RExt-8.1): https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-15.0%2BRExt-8.1

[26] Yu, H.; Cohen, R.; Rapaka, K.; Xu, J.: Common Conditions for Screen Content Coding Tests. JCTVC-S1015, October 2014.

[27] Bjontegaard, G.: Calculation of Average PSNR Differences between RD-Curves. VCEG-M33, April 2001.

[28] Li, B.; Xu, J.: Comparison of Compression Performance of HEVC Test Model 16.2 and HEVC Screen Content Coding Extensions Test Model 3 with AVC High 4:4:4 Predictive profile. JCTVC-T0042, February 2015.

[29] Lai, P.; Liu, S.; Lei, S.: AHG6: Coding Performance of SCC Tools and Suggested Common Test Conditions. JCTVC-T0080, February 2015.

[30] François, E.; Tabatabai, A.; Alshina, E.: BoG Report: Methodology for Evaluating Complexity of Combined and Residual Prediction Methods in SHVC. JCTVC-L0440, January 2013.

[31] Lai, P.; Liu, S.; Lei, S.: AHG6: On Adaptive Color Transform (ACT) in SCM2.0. JCTVC-S0100, October 2014.

[32] Xu, X.; Liu, S.; Lei, S.: On Unification of Intra Block Copy and Inter-Picture Motion Compensation. JCTVC-Q0132, April 2014.

[33] Li, B.; Xu, J.: Non-SCCE1: Unification of Intra BC and Inter Modes. JCTVC-R0100, July 2014.

[34] Li, B.; Xu, J.; Xu, X.; Liu, S.; Lei, S.: CE2: Result of Test 1. JCTVC-S0080, October 2014.

[35] Pang, C. *et al.*: Non-CE2: Intra Block Copy and Inter Signalling Unification. JCTVC-T0227, February 2015.

[36] Ye, J.; Liu, S.; Lei, S.: CE1-Related: Palette Coding for Non-444 Format Content. JCTVC-T0072, February 2015.

[37] Joshi, R.; Pu, W.; Seregin, V.; Karczewicz, M.; Zou, F.: Non-CE1: Extension of Palette Mode to Non-4:4:4 Formats. JCTVC-T0109, February 2015.

[38] Xiu, X.; Ye, Y.; He, Y.: Palette Coding Mode for Non-444 Screen Content Video. JCTVC-T0120, February 2015.

[39] Chang, T.S.; Chen, C.C.; Liao, R.L.; Kuo, C.W.; Peng, W.H.: Intra line copy for HEVC screen content coding, in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conf. (APSIPA ASC-2014)*, Cambodia, December 2014.

[40] Zhao, L.; Chen, X.; Lin, T.: Mapping of intra string copy parameters for HEVC screen content coding, in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conf. (APSIPA ASC-2014)*, Cambodia, December 2014.

[41] Chen, C.-C.; Chen, Y.; Xu, J.; Lin, T.; Wang, W.: CE3: Summary Report for Core Experiment 3 on Intra Line Copy and Intra String Copy. JCTVC-T0023, February 2015.

Dr. Shan Liu has been with MediaTek (USA) Inc. since 2010, where she currently serves as the Director of Technology development. While at MediaTek, Shan and her team made a significant amount of contributions to the High-Efficient Video Coding (HEVC/H.265) standard and its extensions, among which more than 40 proposed technologies have been adopted into the standards. Prior to joining MediaTek, Shan worked for a Silicon Valley start-up, SiBEAM Inc. which was acquired by Silicon Image in 2010. From 2006 to 2008, Shan was a principal member of research staff at Mitsubishi Electric Research Laboratories (MERL) in Cambridge, MA. During her stay at MERL, Shan led several industrial and government projects; she also conducted future research in video and image coding and represented the company in international standardization activities. From 2003 to 2006, Shan held Senior Engineering positions at Sony Electronics and Sony Computer Entertainment of America, where she developed the PlayStation3 Blu-ray decoder with her team. Previously, Shan also held senior and interim technical positions at Samsung Electronics, IBM T.J. Watson, Rockwell Science Center, InterVideo Inc. and AT&T Labs, Research.

Dr. Shan Liu received her M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, Los Angeles, CA, and the B.E. degree from Tsinghua University, Beijing, China. She has been granted more than 20 US and global patents and has numerous patent applications in process. She has published more than 30 peer-reviewed technical papers and more than 200 international standard contributions. She is a senior member of IEEE.

Xiaozhong Xu received his B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, and the MS degree from Polytechnic School of Engineering, New York University, NY, all in Electrical Engineering. He is now with MediaTek (USA) as a member of technical staff. Prior to joining MediaTek, he worked for Zenverge, Inc., a semiconductor company working on multi-channel video transcoding ASIC design. He also held technical positions at Thomson Corporate Research (now Technicolor) and Mitsubishi Electric Research Laboratories. Dr. Xu is an active participant in video coding standardization activities. He has successfully contributed to various standards, including H.264/AVC, AVS (China), and HEVC Screen Content Coding extensions.

Shaw-Min Lei (S'87-M'88-SM'95-F'06) received his B.S. and M.S. degrees from the National Taiwan University, Taipei, Republic of China, in 1980 and 1982, respectively, and the Ph.D. degree from the University of California, Los Angeles in 1988, all in Electrical Engineering. From August 1988 to October 1995, he was with Bellcore (Bell Communications Research), Red Bank, New Jersey, where he had worked mostly in video compression and communication areas and for a short period of time in wireless communication areas. From October 1995 to March 2007, he was with Sharp Laboratories of America, Camas, Washington, where he was a manager of the Video Coding and Communication Group. Since March 2007, he has been with MediaTek, Hsinchu, Taiwan, as a Director of Multimedia Technology Division, working in video/image coding/processing, computer vision, acoustics/speech processing, and bio-medical signal processing areas. His group has made a significant amount of contributions to the High-Efficient Video Coding (HEVC or H.265) standard. Under his direction, his group has become one of the top contributors in the video coding standard bodies, ISO MPEG, and ITU-T VCEG. His research interests include video/image compression, processing and communication, picture quality enhancement, computer vision, and digital signal processing. He has published more than 90 peer-reviewed technical papers and more than 550 contributions to MPEG4, JPEG2000, H.263+, H.264, and HEVC international standard meetings. He has been awarded more than 75 patents. He is an IEEE fellow.

Dr. Yucheun Kevin Jou is a Senior Corporate Vice President and Chief Technology Officer at MediaTek Inc., a leading fabless semiconductor company in the world with products in cellular phones, tablets, wireless connectivity, home entertainment, and optical storage. As the Chief Technology Officer, he provides guidance to the company's technology and business strategies, and is responsible for advanced technology research and development in the company. Additionally, he oversees engineering teams for communication system design, multimedia, and computing system engineering. He is also responsible for sponsored university researches and other joint programs with external research institutes and government agencies. Before joining MediaTek in 2011, Dr. Jou spent nearly 22 years at Qualcomm Incorporated. He was involved in the design and development of the original CDMA prototype system, the IS-95 standard, and several early CDMA base station and mobile station modem chips. He was a key contributor to the design and standardization of the third generation (3G) cellular systems, including leading the development of CDMA2000 standards for voice and packet data services. In particular, Dr. Jou was innovative in applying interference cancellation techniques and intelligent signal transmission to wireless voice communications, which resulted in a system with industry-leading voice capacity up to this date. He was also involved in the design of the Globalstar LEO satellite communication system. Dr. Jou played a major role in Qualcomm's technical and business activities in the Greater China area. He served as Qualcomm China's Chief Technology Officer from 2003 to 2005. Dr. Jou holds approximately 80 US patents, many of which are used in all 3G cellular systems. He has published numerous technical papers in the wireless communications area. Dr. Jou received a Bachelor of Science degree in Electrical Engineering from National Taiwan University in 1982 and Master of Science and Ph. D. degrees, both in Electrical Engineering, from the University of Southern California in 1985 and 1989, respectively.