IMPROVED PALETTE INDEX MAP CODING ON HEVC SCC

Yu-Chen Sun, Tzu-Der Chuang, Jungsun Kim, Yi-Wen Chen, Shan Liu, Yu-Wen Huang, Shawmin Lei

MediaTek Inc.

ABSTRACT

Palette mode is the new coding tool that has been adopted in the Screen Content Coding Extensions of High Efficiency Video Coding (HEVC SCC). Palette mode can represent colour clusters for screen content efficiently and can be summarized into two parts: palette coding tools and colour index map coding tools. This paper proposes two techniques to improve colour index map coding: transition copy and prediction across coding unit boundary. The former is for exploring the correlation between an index and all of the previously coded indices; the latter for utilizing the correlation between the content along the coding unit boundary. Experimental results reportedly show that, compared with HEVC SCC, the proposed techniques can achieve 3.2%, 2.2%, and 1.7% BD-rate savings compared with HEVC SCC for "YUV, text & graphics with motion, 1080p & 720p sequences" under all intra, random access, and low-delay B common test conditions, respectively.

Index Terms**—** Screen Content Coding, High Efficiency Video Coding, HEVC, HEVC SCC, Palette Mode

1. INTRODUCTION

In recent years, screen content coding has been widely used for many applications, such as remote desktop, web conferencing, and cloud computing [1]. In March 2014, the Joint Collaborative Team on Video Coding (JCT-VC), formed by ISO/IEC MPEG and ITU-T VCEG, started to standardize Screen Content Coding Extensions of High Efficiency Video Coding (HEVC SCC) [2]. Palette mode [3] is one of the major coding tools adopted in HEVC SCC and can efficiently describe all pixels in a coding unit (CU) with few selected representative colours and demonstrate impressive bit-rate saving.

Palette mode is designed according to the observation that the pixel values in a screen content frame usually concentrate on few colour values. Figure 1 shows an example of the palette mode. The encoder analyzes the pixels in a CU and determines several representative colours to construct a palette; i.e., a colour mapping table between the representative colour values and colour indices. The

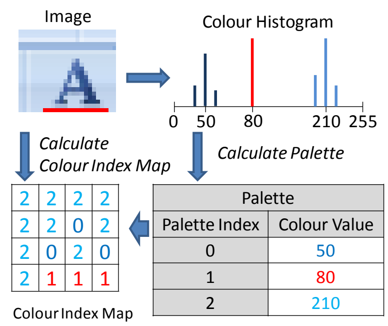
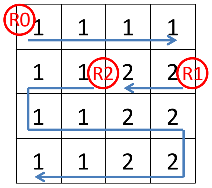


Fig. 1. An example of palette mode

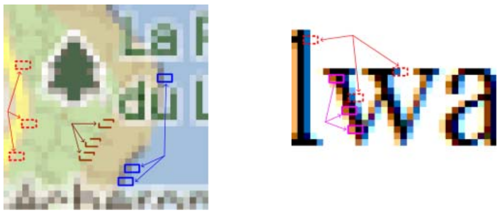
palette is signalled in the bitstream. The pixels with the pixel values close to the palette colours are quantized to form the palette colours and represented by the corresponding colour indices. The rest of the pixels are called escape pixels. The pixel values of the escape pixels are signalled directly. A special colour index value is reserved to represent the escape pixels. All colour indices in the CU form a colour index map, which is transmitted to the decoder along with the escape pixel values. Note that in Fig. 1, to simplify the illustration, a pixel or a colour index is shown to correspond to only one value. However, in HEVC SCC Draft [2], a pixel or a colour index could represent three color component values (e.g., YCbCr or GBR).

Palette mode has been investigated for several years [3] -[8]. During HEVC SCC standardization, several palette mode methods have been proposed [3][6]-[8] and can be summarized into two parts: palette coding tools and colour index map coding tools. In HEVC SCC, a palette is predictively coded and an index map is coded by a run-based approach. The palette coding methods have been intensively studied in the past years [9]. Several tools, e.g., prediction based on the last coded palette [10], palette predictor stuffing [11][12], etc., have been proposed to improve the palette coding efficiency by exploring the correlation between a palette and its previous coded palettes. A sophisticated palette coding method has been adopted in HEVC SCC [9]. Because the index map coding method is rarely discussed, it is relatively rudimentary. It still has room for improvement. For instance, in HEVC SCC, the colour indices can only be predicted by other indices within the same CU. However, the pixel values of a current CU are highly correlated to those of its neighboring boundary pixels.



Run Index	Run Mode	Palette Index (derived from palette_index)	Run of Pixels (derived from PaletteIndexRun)
R0	Copy Index	1	4
R1	Copy Index	2	2
R2	Copy Above	N/A	10

Fig. 2. An example of colour index map coding.



(a) (b)

Fig. 3. An example of repeating color transition patterns in screen content.

If this correlation can be explored, significant improvements can be obtained. In this paper, a new technique, prediction across CU boundary, is proposed to address this issue. In addition, HEVC SCC only allows an index to be predicted by its above index. Yet, we have found that an index correlates with not only its above index, but also with all of its neighboring indices, as well as the previous coded indices with similar neighboring indices. To address this issue, a transition copy (TC) technique is proposed.

This paper is organized as the follows. In section 2, an overview of the palette mode in HEVC SCC is introduced, and then two issues in HEVC SCC are addressed. Two new coding tools, Transition Copy (TC) and Prediction across CU Boundary, are proposed to address the issues in section 3. In Section 4, the experimental results are shown to demonstrate the coding gains of individual techniques and a combination of two proposed techniques. Section 5 concludes this paper.

2. PALETTE MODE OVERVIEW

In HEVC SCC palette mode, a flag is transmitted at the CU level to signal whether the palette mode is used. If the palette mode is utilized, pixels in a CU are represented by major colours and escape colours. Major colours are representative colours in the CU. Escape colours are the colours that do not belong to the major colours. Note that a colour is a 3-value (e.g., YCbCr or GBR) vector.

Major colours are described by a palette table, and encoded by palette table coding tools. Based on the palette table, a pixel of three colour components is represented by a palette index. Palette indices of all pixels in a CU form a palette index map and are encoded by palette index map coding tools.

Predictive coding is applied for the palette table. The decoder constructs a palette table predictor to predict the palette table of the current CU [9]-[12]. After the palette table is encoded, the pixels in a CU are represented by palette indices corresponding to the colours in the palette. The indices form an index map, are divided into several runs and then encoded in horizontal or vertical traverse scan order [14][15]. In this paper, the horizontal scan order is assumed in the examples for illustration. Figure 2 shows an example of the palette index map coding. In the figure, an index map of 16 indices is encoded by three runs with horizontal traverse scan. There are two run modes: copy index mode and copy above mode. For each starting position of a run, a flag is transmitted to indicate which run mode is used.

If the copy above mode is used, a piece of PaletteIndexRun information is transmitted. A run of pixels copy the palette indices from their above pixels, where the run value is derived from PaletteIndexRun. For instance, in Fig.2, R2 is a copy above runs, whose run value is 10.

If the copy index mode is used, a **palette_index_idc** syntax element is first signalled, followed by a piece of PaletteIndexRun information. A run of pixels share the same palette index, where the palette index and the run value are derived from the **palette_index_idc** and PaletteIndexRun, respectively. For instance, in Fig.2, R0 and R1 are two copy index runs, whose (palette index, run value) are (1, 4) and (2, 2), respectively.

Over the past few years, palette coding has been extensively studied [9]. However, in comparison with palette table coding, palette index map coding is relatively less optimized and can be further improved. In the following subsections, two techniques are presented: transition copy and prediction across CU boundary. The former one can improve the performance of copy index mode, while the latter one can enhance the copy above mode.

3. IMPROVEMENTS OF PALETTE INDEX CODING

3.1. Transition Copy

In screen content, the occurrence of a colour usually correlates with its neighboring colours. Figure 3 illustrate this phenomenon using two examples. In Fig. 3(a), several repeating neighboring colour pairs can be observed in the object boundary. For example, a colour transition between gray and blue frequently occurs at the border between the land and sea. They are marked by the blue squares in Fig. 3(a). Other types of colour transition patterns are also indicated by squares of different colours in Fig. 3(a). Besides, sub-pixel anti-alias rendering [16] is usually applied to the text region in screen content. This generates more repeating colour transition patterns, such as the examples shown in Figure 3(b): the red colour often precedes the black colour, and the blue colour is usually near the white colour.

Based on the above observation, a new colour index prediction method, transition copy (TC), is proposed. TC is

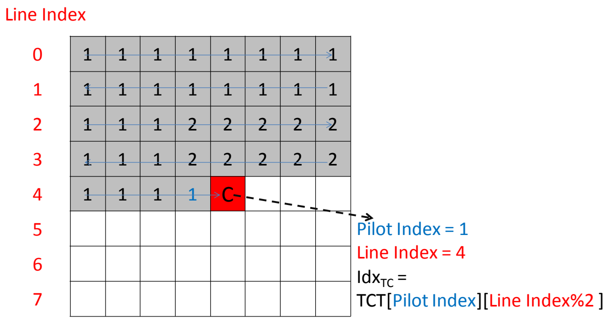


Fig. 4. An example of the transition copy (TC) prediction method.

TABLE I AN EXAMPLE OF TC TABLE

Pilot Index	TC Predictor #0	TC Predictor #1
0	TCT[0][0]	TCT[0][1]
…	…	…
N_{TC}	TCT[N_{TC}][0]	TCT[N_{TC}][1]

originally introduced in [17] and subsequently improved in [18]. It utilizes the transition correlation between a colour and its neighboring colours to further increase coding efficiency. TC is an additional colour index value prediction method for the copy index run mode. If an index is encoded by a copy index run, prior to encoding the index value, the encoder will derive a new index predictor, denoted as Idx_{TC}, using the TC method. If the prediction hits, the index value of the copy index run is inferred as Idx_{TC}, and the bits for the encoding index value (i.e., **palette_index_idc**) can be saved. If the prediction is unsuccessful, the index value will be encoded.

An example of the TC palette index predictor is presented in Fig. 4. The current pixel, C, has a pilot palette index of the value 1. The shaded squares labeled with numbers are coded pixels. From the coded pixels, it can be found that when a palette index, 1, is coded in a left-to-right direction, the next palette index to the right is likely to be 2. This transition pattern (i.e., index 1=> index 2) is recorded in the TC table to enhance the efficiency of palette index prediction. This is the concept of TC.

When the current CU is the first CU of the slice, a TC table is reset at the first pixel. However, when the current CU is a non-first CU of the slice, the TC table inherits that of the previous CU. We call this inheriting property the "TC table propagation." The TC table is maintained and updated as the pixels in the CU are being coded. As shown in Table I, for each non-first-column pixel that is to be coded, the input of the TC table is called a pilot palette index, Idx_{pilot}, which is set to the left or right of the previous pixel depending on the scan direction. In addition, the output of the TC table is the TC palette index, Idx_{TC}. For a pilot palette index, Idx_{pilot}, two TC palette indices, TCT[Idx_{pilot}][0] and TCT[Idx_{pilot}][1], are recorded in the TC table. Whereas TCT[Idx_{pilot}][0] is likely to occur at a pixel position to the right of the pilot palette index relative to the causal area of the CU, TCT[Idx_{pilot}][1] is likely to occur at a pixel position to the left of the pilot palette index. A pixel on the N_{th} row will choose TCT[Idx_{pilot}][N_{th}%2] as the TC predictor.

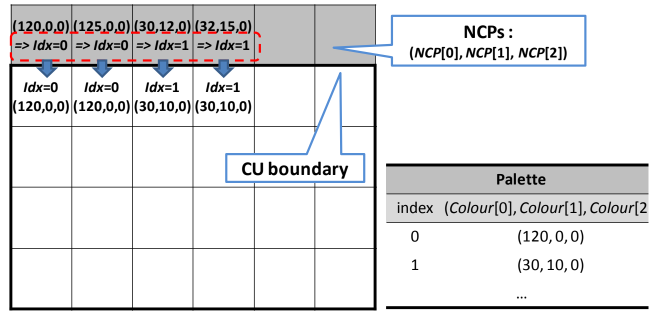


Fig. 5. An example of prediction across CU boundary.

The TC predictor will be used to predict the current palette index. If the prediction misses, the pilot index and the current index, (Idx_{pilot}, the current palette index), become a new transition pattern that updates the TC table. In addition, the current index is sometimes found to be equal to its pilot index. Such a pattern is inefficient. Therefore, we only update the TC table if the current index is not the same as the pilot colour index. To update a TC table, the encoder and decoder replace TCT[Idx_{pilot}][N_{th}%2] with the current palette index and TCT[the current palette index][(N_{th} +1)%2], with Idx_{pilot}. The two updated values correspond to the same transition pattern yet differ in the scan directions.

3.2. Prediction across CU Boundary

In HEVC-SCC, the run of the first column cannot be a copy above run because the above pixel position is outside the current CU and no corresponding indices can be used for prediction in a copy above run. Moreover, the run type is not signalled and forced to be copy index run. In this scheme, the correlation between the current CU and a neighboring CU remains unused. If the correlation can be further utilized, coding efficiency can be improved.

In this subsection, we propose to use the partially decoded pixels (i.e., before deblocking) of the nearest above row (or the nearest left column depending on the scan order) in a neighboring CU to generate the index prediction sources [19]-[21]. These pixels are referred to as the neighboring CU pixels (NCPs). For example, Fig. 5 shows the NCPs marked in gray. An NCP, NCP_i, has three decoded pixel values, ($NCP_i[0]$, $NCP_i[1]$, $NCP_i[2]$), corresponding to YCbCr or GBR.

To reconstruct the palette indices of the current CU, the decoder transforms each NCP's values into a palette index that minimizes the difference between the NCP's values and the pixel values corresponding to the palette index in the current CU's palette table. Assume that there are N colours in the palette table, e.g. $Colour_0$, $Colour_1$, … $Colour_{N-1}$, where a colour, $Colour_j$, has three pixel values, ($Colour_j[0]$, $Colour_j[1]$, $Colour_j[2]$), and a palette index j. For an NCP, NCP_i, its transformed index, Idx_i, is defined as:

$$Idx_i = \arg\min_{0\le n<N}\left(\sum_{k=0}^{2}\left|NCP_i[k]-Colour_n[k]\right|\right). \tag{1}$$

TABLE II. BD-RATE PERFORMANCE OF EACH INDIVIDUAL PALETTE IMPROVEMENT. THE ANCHOR IS SCM-4.0. THE COLUMNS, "(1) TRANSITION COPY", "(2) PREDICTION ACROSS CU", AND "(1)+(2)", PRESENT SCM-4.0 COMBINING TRANSITION COPY, PREDICTION ACROSS CU, AND ALL OF PALETTE IMPROVEMENTS, RESPECTIVELY. NEGATIVE VALUES MEAN BIT RATE DECREASES (CODING GAINS) AND CAN REFLECT THE BENEFITS OF EACH INDIVIDUAL PALETTE IMPROVEMENT. THE FULL FRAME SEARCH CONFIGURATION FOR IBC IS USED.

	(1) Transition Copy			(2) Prediction across CU			(1) + (2)		
	AI	RA	LB	AI	RA	LB	AI	RA	LB
RGB, text & graphics with motion, 1080p & 720p	-1.0%	-0.7%	-0.5%	-1.8%	-1.3%	-1.1%	-2.8%	-1.9%	-1.5%
RGB, mixed content, 1440p & 1080p	-0.2%	-0.1%	-0.3%	-0.8%	-0.5%	-0.2%	-0.9%	-0.7%	-0.5%
RGB, Animation, 720p	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	-0.1%
RGB, camera captured, 1080p	0.0%	0.0%	-0.1%	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%
YUV, text & graphics with motion, 1080p & 720p	-0.9%	-0.6%	-0.4%	-2.4%	-1.6%	-1.3%	-3.2%	-2.2%	-1.7%
YUV, mixed content, 1440p & 1080p	-0.3%	-0.2%	-0.2%	-1.0%	-0.8%	-0.5%	-1.3%	-0.9%	-0.8%
YUV, Animation, 720p	0.0%	0.0%	0.1%	-0.1%	-0.1%	-0.2%	-0.1%	0.0%	-0.1%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	105%	104%	103%	106%	101%	101%	112%	106%	105%
Dec Time[%]	104%	105%	104%	103%	103%	104%	105%	105%	105%

Figure 5 shows an example of the palette of the current CU. The decoder uses the palette to transform the NCP's values into palette indices. The resulting indices, *Idx*, of the NCPs from Eq.(1) are listed underneath the pixel values in Fig.5. The copy-above run mode at the CU boundary is also illustrated according to the proposed scheme. To reconstruct a palette index for the first row of the current CU, if the index is encoded by a copy-above run mode, the decoder reconstructs the index from the corresponding NCP index.

In the aforementioned method, the computation burden might be increased if the decoder is required to transform the NCP's partially decoded sample values into a palette index by Eq.(1). As a solution, instead of deriving an index from the NCP, we directly use the NCP's partially decoded pixel values to reconstruct pixels in the current CU. Therefore, the conversion process, Eq. (1), can be removed entirely. To reconstruct the sample values of the first row of the current CU, if these pixels' indices are encoded by a copy-above run, the decoder reconstructs the pixels from the NCP pixel values instead of the NCP-derived indices. The results in [21] show that the coding loss of the simplification is minor and indicate that the simplified method achieves good performance-complexity trade-off.

4. EXPERIMENTAL RESULTS

The proposed improvements have been integrated into the HEVC SCC reference software, SCM-4.0[22]. In SCM-4.0, the syntax element of palette indices is proposed to be grouped and moved in front of the run mode and run length syntax [23]. However, after the transition copy and the simplified prediction across CU boundary method are introduced, the run mode and run length information is required to parse the palette index syntax. To accommodate the proposed palette techniques, the syntax grouping in [23] is removed, which does not have any coding efficiency impact. The experiments are conducted under the common test conditions defined by JCT-VC [24]. Intra block copy mode is enabled, and the full frame search configuration is used. The coding performance was measured by the BD-rate [25]. To simplify the result tables, in the section, only BD-rate results of Y/G component are shown. The results of other two components are similar, so they are omitted for brevity. The simulations are performed using the 64bit Linux platform with Xeon 5160 3.0GHz CPUs.

The results are shown in the Table II. For "YUV, text & graphics with motion, 1080p & 720p sequences" under all intra (AI) configuration, 0.9% and 2.4% BD-rate saving could be observed for (1) transition copy and (2) prediction across CU boundary, respectively, while the combination of the two palette improvements can improve coding efficiency by 3.2%. Regarding the implementation complexity, prediction across CU boundary allows the pixel reconstruction process of the palette mode to access pixels outside of the CU as well as the intra prediction module. The buffer of those pixels is shared with the intra prediction module. Transition copy requires the decoder to maintain a TC table which is 310 bits in this paper. The operation is quite simple, including looking up the table to find the TC predictor and updating it. The run time information is also summarized in TABLE II, and it can be found that the decoding run time increase are within 5% while the encoding time increases are 12%, 6%, and 5% under AI, RA, and LB configurations, respectively.

5. CONCLUSION

In this paper, two techniques are proposed to improve the palette map coding of the palette mode in HEVC SCC, including transition copy and prediction across CU boundary. Experimental results show that compared with HEVC SCC, the proposed techniques can achieve up to 3.2% BD-rate savings under the all intra test condition. Since screen content coding is used in many applications, improvements are desirable and should be investigated in the future. In addition to our approaches, strategies using the string matching method have been introduced in [26] and [27] to help enhance palette map coding. Combining our techniques with these string matching methods may be a promising direction for improving the current HEVC SCC and future codec development.

6. REFERENCES

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

[2] R. Joshi, S. Liu, G. J. Sullivan, J. Xu, Y. Ye, "HEVC Screen Content Coding Draft Text 4," in Document of Joint Collaborative Team on Video Coding, JCTVC-U1005, 21st Meeting, Warsaw, PL, 19 – 26 June 2015.

[3] P. Onno, X. Xiu, Y.-W. Huang, R. Joshi "Suggested combined software and text for run-based palette mode," in Document of Joint Collaborative Team on Video Coding, JCTVC-R0348, 18th Meeting, Sapporo, JP, 30 June – 9 July 2014.

[4] C. Lan, G. Shi, and F. Wu, "Compress Compound Images in H.264/MPGE-4 AVC by Exploiting Spatial Correlation," in *IEEE Trans. on Image Processing*, Vol. 19, No. 4, pp. 946-957, Apr. 2010.

[5] Z. Pan, H. Shen, Y. Lu, S. Li, and N. Yu, "A low-complexity screen compression scheme for interactive screen sharing," in *IEEE Trans. on Circuits Systems for Video Technology*, Vol. 23, No. 6, pp. 949-960, Jun. 2013.

[6] X. Guo, Y. Lu, and S. Li, "RCE4 Test1: Major-color-based screen content coding," in Document of Joint Collaborative Team on Video Coding, JCTVC-P0108, 16th Meeting, San Jose, US, 9-17 Jan. 2014.

[7] L. Guo, W. Pu, M. Karczewicz, J. Sole, R. Joshi, and F. Zou, "RCE4: Results of Test 2 on palette mode for screen content coding," in Document of Joint Collaborative Team on Video Coding, JCTVC-P0198, 16th Meeting, San Jose, US, 9-17 Jan. 2014.

[8] Y.-C. Sun, T.-D. Chuang, P. Lai, S.-T. Hsiang, Y.-W. Chen, X. Zhang, S. Liu, Y.-W. Huang, and S. Lei, "AHG10: A triplet palette mode combining JCTVC-P0108 and JCTVC-P0198," in Document of Joint Collaborative Team on Video Coding, JCTVC-Q0083, 17th Meeting, Valencia, ES, 27 Mar. – 4 Apr. 2014.

[9] Y.-C. Sun, T.-D. Chuang, P. Lai, Y.-W. Chen, S. Liu, Y.-W. Huang S. Lei, "Palette Mode - a New Coding Tool in Screen Content Coding Extensions of HEVC," in *Proc. IEEE Int. Conf. Image Process(ICIP)*, Sep. 2015.

[10] P. Lai, S. Liu, T.-D. Chuang, Y.-W. Huang, and S. Lei, "Non-RCE4: Major color table (palette) sharing," in Document of Joint Collaborative Team on Video Coding, JCTVC-P0153, 16th Meeting, San Jose, US, 9-17 Jan. 2014.

[11] Y.-C. Sun, P. Lai, T.-D. Chuang, Y.-W. Chen,S. Liu, Y.-W. Huang, and S. Lei, "SCCE3 Test A.2: Palette coding and palette predictor updates," in Document of Joint Collaborative Team on Video Coding, JCTVC-R0119, 18th Meeting, Sapporo, JP, 30 June – 9 July 2014.

[12] C. Gisquet, G. Laroche, and P. Onno, "SCCE3: Test A.3 - Palette stuffing," in Document of Joint Collaborative Team on Video Coding, JCTVC-R0348, 18th Meeting, Sapporo, JP, 30 June – 9 July 2014.

[13] B. Li, J. Xu, and G. J. Sullivan, "Comparison of Compression Performance of HEVC Screen Content Coding Extensions Test Model 5 with AVC High 4:4:4 Predictive profile" in Document of Joint Collaborative Team on Video Coding, JCTVC-V0033, 22th Meeting, Geneva, CH, 15–21 Oct. 2015.

[14] Y.-C. Sun, T.-D. Chuang, Y.-W. Chen, X. Zhang, Y.-W. Huang, and S. Lei, "SCCE3 Test B.4: Adaptive color index map scan," in Document of Joint Collaborative Team on Video Coding, JCTVC-R0057, 18th Meeting, Sapporo, JP, 30 June – 9 July 2014

[15] C.-M. Tsai, Y. He, X. Xiu, and Y. Ye, "SCCE3: Test B.9 - BWT-based index grouping," in Document of Joint Collaborative Team on Video Coding, JCTVC-R0168, 18th Meeting, Sapporo, JP, 30 June – 9 July 2014.

[16] C. Betrisey, J. F. Blinn, B. Dresevic, B. Hill, G. Hitchcock, B. Keely, and T. Whitted, "20. 4: Displaced Filtering for Patterned Displays," in *SID Symposium Digest of Technical Papers*, Vol. 31, No. 1, p.p. 296, May 2000.

[17] C. Gisquet, G. Laroche, and P. Onno, "Non-RCE4: Transition copy mode for Palette mode," in Document of Joint Collaborative Team on Video Coding, JCTVC-P0115, 16th Meeting, San Jose, US, 9-17 Jan. 2014.

[18] Y.-C. Sun, T.-D. Chuang, Y.-W. Chen, Y.-W. Huang, and S. Lei, "CE6 Test C.2: Transition copy mode," in Document of Joint Collaborative Team on Video Coding, JCTVC-S0078, 19th Meeting, Strasbourg, FR, 17 – 24 Oct. 2014.

[19] Y.-C. Sun, T.-D. Chuang, Y.-W. Chen, Y.-W. Huang, and S. Lei, "Non-RCE4: Cross-CU major color index prediction," in Document of Joint Collaborative Team on Video Coding, JCTVC-P0093, 16th Meeting, San Jose, US, 9-17 Jan. 2014.

[20] Y.-C. Sun, J. Kim, T.-D. Chuang, Y.-W. Chen, S. Liu, Y.-W. Huang, and S. Lei, "Non-CE6: Cross-CU palette colour index prediction," in Document of Joint Collaborative Team on Video Coding, JCTVC-S0079, 19th Meeting, Strasbourg, FR, 17 – 24 Oct. 2014.

[21] Y.-C. Sun, J. Kim, S. Liu, T.-D. Chuang, Y.-W. Chen, Y.-W. Huang, S. Lei, V. Seregin, F. Zou, W. Pu, R. Joshi, and M. Karczewicz, "CE1: Test A.1: extended copy above mode to the first line (1.1-1.5)," in Document of Joint Collaborative Team on Video Coding, JCTVC-T0036, 20th Meeting, Geneva, CH, 10 – 18 Feb. 2015.

[22] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.2+SCM-4.0/

[23] M. Karczewicz, W. Pu, R. Joshi, and V. Seregin, "Non CE1: Grouping Palette Indices At Front," in Document of Joint Collaborative Team on Video Coding, JCTVC-T0065, 20th Meeting, Geneva, CH, 10–18 Feb. 2015.

[24] H. Yu, R. Cohen, K. Rapaka, and J. Xu, "Common test conditions for screen content coding," in Document of Joint Collaborative Team on Video Coding, JCTVC-T1015, 20th Meeting, Geneva, CH, 10–18 Feb. 2015.

[25] G. Bjøntegaard, "Improvements of the BD-PSNR Model", in Document VCEG-AI11, ITU-T SG16, Jul. 2008

[26] Z. Ma, W. Wang, M. Xu, and H. Yu, "Advanced screen content coding using color table and index map," in *IEEE Trans. on Image Processing* , Vol. 23, issue 10, pp. 4399-4412, Oct. 2014.

[27] Y. Xu, W. Huang, W. Wang, F. Duanmu, and Z. Ma, "2-D Index Map Coding for HEVC Screen Content Compression," *in Proc. IEEE Data Compression Conference (DCC)*, pp. 263-272, Apr. 2015.