PALETTE MODE – A NEW CODING TOOL IN SCREEN CONTENT CODING EXTENSIONS OF HEVC

Yu-Chen Sun[*], *Tzu-Der Chuang*[*], *PoLin Lai*[+], *Yi-Wen Chen*[*], *Shan Liu*[+], *Yu-Wen Huang*[*], *Shawmin Lei*[*]

[*]MediaTek Inc.
No.1, Dusing 1st Rd., Hsinchu Science
Park, Hsinchu City 30078, Taiwan

[+]MediaTek USA Inc.
2860 Junction Ave,
San Jose, CA 95134, USA

ABSTRACT

Palette mode, as a new coding tool, is adopted in the Screen Content Coding Extensions of High Efficiency Video Coding (HEVC SCC) that is being defined by the Joint Collaborative Team on Video Coding (JCT-VC). In the palette mode, pixels in a coding unit (CU) are represented by selected representative colours according to the characteristics of screen contents in which pixel values usually concentrate on few colour values. This paper introduces the palette mode in HEVC SCC, and our contributions to the palette mode are also presented. Experimental results show that disabling the palette mode suffers 20.3%, 13.0%, and 7.5% BD-rate increases for "YUV, text & graphics with motion, 1080p & 720p sequences" under all intra (AI), random access (RA), and low-delay B (LB) common test conditions, respectively.

***Index Terms*—** palette, screen content coding, SCC, High Efficiency Video Coding, HEVC

1. INTRODUCTION

In recent years, screen content coding has been widely used for many applications, such as remote desktop, web conferencing, and online training. In March 2014, the Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC MPEG and ITU-T VCEG started the standardization of Screen Content Coding Extensions of High Efficiency Video Coding (HEVC SCC). Three major coding tools are newly adopted [1]: intra block copy, adaptive colour transform, and palette mode. This paper focuses on palette mode.

Palette mode is designed according to the observation that pixel values in screen contents usually concentrate on few colour values. Fig.1 shows an example of palette mode. The encoder analyzes pixels in a coding unit (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 palette is transmitted to the decoder. Based on the palette, most pixels in the CU that are close to palette colours are quantized to palette colours and represented by colour indices. The rest of the pixels that

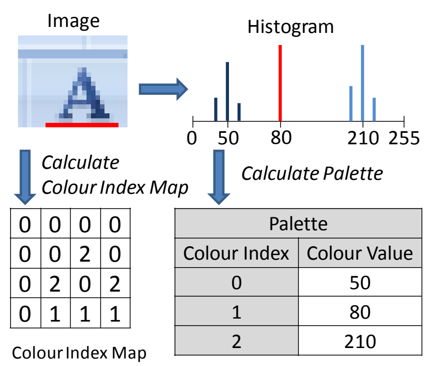


Fig. 1. An example of palette mode

are far from any palette colours will not be quantized the to palette colour value. Instead, their pixel values will be signalled directly. These pixels are called escape pixels. A special colour index value is reserved for representing escape pixels. All the colour indices in the CU form a colour index map, and the map is transmitted to the decoder together with the escape pixel values. Please note that in Fig. 1, to simplify the illustration, a pixel or a colour index corresponds to one value. However, in HEVC SCC Draft 2 [1], a pixel or a colour index corresponds to three color component values (e.g., YCbCr or GBR).

Palette mode has been investigated for several years [2][3]. Several palette mode methods were proposed [4]-[7] and can be summarized into two parts: palette coding tools and colour index map coding tools. A palette is constructed and predictive encoded by palette coding tools. In [4], each colour component of a CU has its own palette, and each sample of a colour component is coded with one sample value index. In [5], three colour components share a triplet palette, and a Y (G) sample and its corresponding Cb (B) and Cr (R) samples are grouped and indicated by one colour index. The colour index is also called palette index, as written in the HEVC SCC text. In [6], a different triplet palette representation is used to describe a colour by three component-wise palette and one triplet palette. To encode a colour index map, in [4], sample value indices are coded sample by sample with options of sample prediction modes, or an entire sample line in the CU are coded with one of two line modes. One line mode copies the previous line entirely, while the other line mode duplicates the first sample for the entire line. In [5], colour indices are coded sample by sample with or without prediction, or multiple samples are coded with a run mode that allows duplicating multiple samples across sample lines. Based on [4], in [6], an improved candidate-based prediction coding method is applied, where the candidate set considers neighboring samples to the left, top, top left, and top right. With pruning of redundant candidates, good coding gain can be achieved. To consider advantages from multiple proposals, a combined palette mode was later proposed in [7]. Compared with the previous individual proposals, the combined palette mode shows significantly higher coding efficiency and even lower complexity. As a result of good collaboration among several companies, the combined palette mode was adopted into the HEVC SCC standard under development in the 18th JCT-VC meeting in July 2014. In following meetings, several palette improvements [9]-[13] were or are being studied on top of [7].

In this paper, the overview of the adopted palette mode in HEVC SCC is introduced in Section 2. Details of several palette techniques are presented in Section 3. In Section 4, the experimental results are shown to demonstrate coding gains of the entire palette mode and individual techniques. Finally, the conclusion and future work are given in Section 5.

2. PALETTE MODE OVERVIEW

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

Major colours are described by a palette and encoded by palette coding tools. Based on the palette, a pixel with three samples of three colour components (e.g., YCbCr or GBR) is represented by a colour index. The colour indices correspond to all pixels in a CU form a colour index map. The index map can be predictive-coded by colour index map coding tools.

2.1. Palette coding tools

The palette is first predicted and then coded. During decoding, the decoder constructs a palette predictor of N_{pred} colours for predicting the palette of the current CU. In [4], the predictor is the palette of a neighboring CU. An improved predictor from previous palette coded CUs is proposed in [6] and [7], which will be introduced in Section 3.1 and Section 3.2.

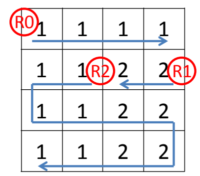
Palette Predictor		Reuse
Colour Index	Colour	Flag
0	$Colour_0$	1
1	$Colour_1$	0
2	$Colour_2$	1
...		0
N_{pred} - 1	$Colour_{N_{pred}-1}$	0

New Colours	
New Colour Size	N_{new}
Colour Index	New Colour
0	$NewColour_0$
1	$NewColour_1$
...	
N_{new} - 1	$NewColour_{N_{new}-1}$

Generate Palette

Current Palette	
Palette Size	N_{new} + 2
Major Colour Index	Major Colour
0	$Colour_0$
1	$Colour_2$
2	$NewColour_0$
...	
N_{new} + 1	$NewColour_{N_{new}+1}$

Fig. 2. An example of palette coding



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

Fig. 3. An example of colour index map coding

Based on the palette predictor, the prediction information of palette is signalled to indicate how to derive the palette of the current CU from the palette predictor. Fig.2 shows an example of palette coding and derivation. In this example, the palette predictor is first created, and N_{pred} reuse flags corresponding to colours in the predictor are signalled. A reuse flag indicates if the colour is reused in the palette of the current CU, and the decoder puts the reused colours (e.g., $Colour_0$ and $Colour_2$ in Fig.2) at the beginning of the palette of the current CU. Next, N_{new} is signalled to derive the number of new colours, and N_{new} new colours, $NewColour_0$… $NewColour_{Nnew-1}$, are signalled. The decoder puts the new colours following the reused colours. In this example, there are (N_{new}+2) colours generated for current palette.

2.2. Colour index map coding tools

Based on the palette, pixels in a CU are represented by colour indices. The indices in the index map are run-length encoded in traverse scan order [8]. There are two run modes: copy index mode and copy above mode. For each starting position of a run, a flag is first transmitted to indicate whether copy index mode or copy above mode is used.

In copy index mode, a palette_index syntax element is first signalled followed by a palette_run syntax element. A run of pixels share a same colour index, where the colour index is derived from the palette_index and the run value is derived from the palette_run.

In copy above mode, only a syntax element palette_run is transmitted. A run of pixels copy the colour indices from their above pixels, where the run value is derived from the palette_run.

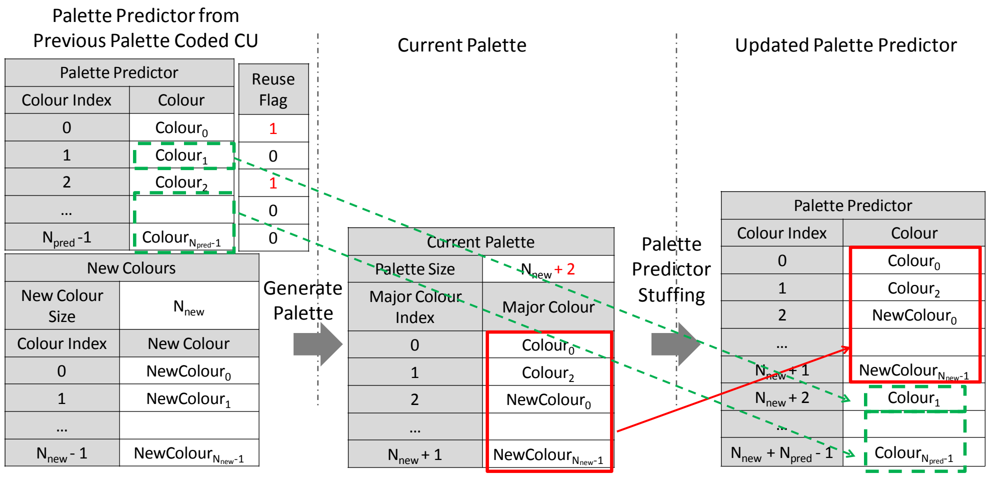


Fig. 4. An example of palette predictor stuffing.

Fig.3 shows an example of colours index map coding. An index map of 16 indices is encoded by three runs with horizontal traverse scan. The first two runs are copy index mode with (colour index, run value) equal to (1, 4) and (2, 2), respectively. The third run is copy above mode with run value equal to 10.

3. PALETTE MODE TECHNIQUES

Based on the palette mode framework described in Section 2, we proposed five techniques to improve the coding efficiency of palette mode. These techniques are included in the HEVC SCC. They are further described in this section.

3.1. Palette predictor propagation

In the palette coding scheme, the decoder needs to maintain a palette predictor. In [4] and [5], the palette predictor is the palette of a neighboring CU. However, if the neighboring CU is not coded using palette mode, there is no valid palette predictor. To solve this issue, propagation of palette predictor information through non-palette CUs was proposed in [6].

We proposed that, during decoding, the decoder constructs a palette predictor of N_{pred} colours for predicting the palette of the current CU. If the current CU is palette coded, the predictor is updated by the palette of the current CU; otherwise, the palette predictor is unchanged and simply propagated to the next CU. In other words, the palette of a CU is predicted by the palette of the last palette coded CU.

3.2. Palette predictor stuffing

Generating palette predictor from the last coded palette, a decoder can always get a valid predictor. However, we observed that a small CU may contain much fewer colours than a large CU. Therefore, the prediction is inefficient when a large CU uses the palette of a small CU as a palette predictor. Based on the observation, we proposed that a decoder can propagate not only the current palette but also the unused colours in the palette predictor to the next CU as the new palette predictor. As a result, after decoding several CUs, the palette predictor can contain sufficient colours for prediction. This property is called as full palette propagation [6] or palette predictor stuffing [14], which can enhance the palette prediction efficiency across different CU sizes.

Fig.4 shows an example of palette predictor stuffing. Compared with using the palette of the last palette coded CU as the predictor (marked by solid line in Fig.4), the unused colours (marked by dashed line in Fig.4) in the previous predictor will be stuffed into the new predictor for the next CU.

3.3. Palette sharing

The colour-by-colour prediction requires colour-wise reuse flags to indicate the prediction operations. However, the major colours may be very similar among CUs in the vicinity, and signalling colour-wise flags might be inefficient in this case. To address this issue, a method of CU-wise palette sharing are proposed [15]. If a palette sharing flag is signalled as 1, a decoder will directly copy all colours from the last coded palette as the palette for the current CU, without signaling several colour-wise reuse flags.

3.4. Index map transposing

In the palette of [4] and [5], the colour indices in a CU are coded in a horizontal scan. However, horizontal scan may not be suitable for all content. Therefore, we proposed to add a vertical scan [16]. Combining traverse scan order [8], a palette_transpose_flag is added in a palette coded CU to select the scan order from horizontal traverse scan and vertical traverse scan.

TABLE I
BD-RATE PERFORMANCE OF THE PALETTE MODE. THE ANCHOR IS SCM-3.0. THE TEST IS SCM-3.0 DISABLING PALETTE MODE. POSITIVE VALUES MEAN BIT RATE INCREASES (CODING LOSSES) AND CAN REFLECT THE BENEFITS OF PALETTE MODE.

	All Intra			Random Access			Low delay B		
	G/Y	B/U	R/V	G/Y	B/U	R/V	G/Y	B/U	R/V
RGB, text & graphics with motion, 1080p & 720p	19.4%	19.6%	19.2%	12.4%	12.9%	12.7%	7.5%	8.5%	8.3%
RGB, mixed content, 1440p & 1080p	3.9%	5.5%	4.9%	2.6%	4.3%	3.8%	1.5%	3.6%	2.8%
RGB, Animation, 720p	0.0%	0.5%	0.6%	0.1%	0.4%	0.3%	0.0%	0.0%	0.0%
RGB, camera captured, 1080p	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	-0.1%
YUV, text & graphics with motion, 1080p & 720p	20.3%	26.9%	29.8%	13.0%	18.8%	22.2%	7.5%	12.9%	15.3%
YUV, mixed content, 1440p & 1080p	6.3%	13.2%	13.6%	4.6%	12.9%	13.0%	2.8%	11.2%	11.4%
YUV, Animation, 720p	-0.1%	2.0%	1.9%	-0.2%	1.4%	2.0%	-0.1%	0.4%	0.7%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%	-0.1%	0.0%	-0.1%	-0.1%	0.0%	0.0%
Enc Time[%]	96%			100%			100%		
Dec Time[%]	104%			99%			98%		

TABLE II
BD-RATE PERFORMANCE OF THE PROPOSED PALETTE TECHNIQUES. THE ANCHOR IS SCM-3.0. THE TEST IS SCM-3.0 DISABLING EACH TECHNIQUE. POSITIVE VALUES MEAN BIT RATE INCREASES (CODING LOSSES) AND CAN REFLECT THE BENEFITS OF THE TECHNIQUES.

	without Palette Predictor Propagation	without Palette Predictor Stuffing	without Palette Sharing	without Index Map Transposing	without Redundancy Removal
RGB, text & graphics with motion, 1080p & 720p	7.3%	4.2%	0.3%	0.5%	0.1%
RGB, mixed content, 1440p & 1080p	2.0%	0.9%	0.0%	0.1%	0.1%
RGB, Animation, 720p	0.1%	0.0%	0.0%	0.0%	0.0%
RGB, camera captured, 1080p	0.0%	0.0%	0.0%	0.0%	0.0%
YUV, text & graphics with motion, 1080p & 720p	7.7%	4.4%	0.5%	0.6%	0.2%
YUV, mixed content, 1440p & 1080p	3.6%	2.2%	0.1%	0.2%	0.1%
YUV, Animation, 720p	0.1%	0.0%	0.0%	0.0%	0.0%
YUV, camera captured, 1080p	0.0%	0.0%	0.0%	0.0%	0.0%

3.5. Removal of redundant copy above runs

To indicate whether the copy index mode or copy above mode is used, a flag is transmitted for each run. However, this syntax design has redundancy when consecutive groups of pixels select the copy above mode. For example, a group of M pixels coded with the copy above mode followed by a group of N pixels coded with the copy-above mode is essentially equivalent to a group of (M+N) pixels coded with the copy above mode. In [17], we proposed to remove the syntax redundancy by adding a constraint as follows: If the copy above mode is chosen by the previous group of pixels, the current group of pixels cannot choose the copy above mode. The mode selection flag is not transmitted in this case, and the current run of pixels is inferred as copy index mode.

4. EXPERMENTAL RESULTS

The palette mode has been integrated into HEVC SCC reference software, SCM-3.0 [18]. In SCM-3.0, palette mode is selected according to rate-distortion evaluation. We evaluate the coding efficiency on top of SCM-3.0. The experiments are conducted under the common test conditions defined by JCT-VC [19]. The coding performance was measured by the BD-rate [20]. The simulations are carried out under 64bit Linux platform with Xeon 5160 3.0GHz CPUs.

4.1 Results of the entire palette mode

To evaluate the palette mode, the test is generated by disabling the palette mode in SCM-3.0, and the anchor is SCM-3.0. The coding losses (i.e., BD-rate increases, positive values) of disabling palette mode can reflect the benefit of the palette mode. As shown in Table I, the BD-rate increases are 20.3%, 13.0%, and 7.5% for "YUV, text & graphics with motion, 1080p & 720p sequences" under AI, RA, and LB, respectively. Runtime-wise there is no significant difference, which indicates a good gain-complexity tradeoff to adopt the palette mode.

4.2 Results of individual palette mode techniques

To evaluate each proposed technique, we use SCM-3.0 (with the palette mode and all techniques) as the anchor and disable each technique. The coding loss (positive values) of disabling each technique can reflect the benefit of the technique. The results are shown in the Table II. Palette predictor propagation and palette predictor stuffing can significantly improve coding efficiency. Palette sharing, index map transposing, and removing redundant copy above runs are somewhat helpful and are adopted due to their simplicities. Note that the index map transposing may have much higher gain in non-common test conditions, as illustrated in [16].

5. CONCLUSION AND FUTURE WORK

In this paper, the palette mode in the HEVC SCC standard under development is introduced. This new tool is designed according to the characteristics of screen contents in which pixel values usually concentrate on few colour values. Experimental results show up to 20.3%, 13.0%, and 7.5% coding gain under AI, RA, and LB, respectively. Future coding gain improvements can be transition copy [9][10] and colour index prediction across CU boundaries [11].

6. REFERENCES

[1] "High Efficiency Video Coding (HEVC) Screen Content Coding: Draft 2," JCTVC-S1005, Oct. 2014.

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

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

[4] X. Guo, Y. Lu, and S. Li, "RCE4 Test1: Major-color-based screen content coding," JCTVC-P0108, Jan. 2014.

[5] 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," JCTVC-P0179, Jan. 2014.

[6] 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," JCTVC-Q0083, Mar. 2014.

[7] P. Onno, X. Xiu, Y.-W. Huang, and R. Joshi, "Suggested combined software and text for run-based palette mode," JCTVC-R0348, June 2014.

[8] C.-M. Tsai, Y. He, X. Xiu, and Y. Ye, "SCCE3: Test B.9 - BWT-based index grouping," JCTVC-R0348, June 2014.

[9] C. Gisguet, G. Laroche, and P. Onno, "CE6: Results of Test C.1 on transition copy mode," JCTVC-S0063, Oct. 2014.

[10] Y.-C. Sun, T.-D. Chuang, Y.-W. Chen, Y.-W. Huang, and S. Lei, "CE6 Test C.2: Transition copy mode," JCTVC-S0078, Oct. 2014.

[11] 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," JCTVC-S0079, Oct. 2014.

[12] K. Zhang, T.-D. Chuang, S. Liu, J. An, X. Zhang, and S. Lei, "CE6-related: Syntax fixes for zero palette in palette coding," JCTVC-S0105, Oct. 2014.

[13] P. Lai, J. Kim, S. Liu, and S. Lei, "Non-CE6: Syntax redundancy removal for palette mode with one index value," JCTVC-S0099, Oct. 2014.

[14] C. Gisquet, G. Laroche, and P. Onno, "SCCE3: Test A.3 - Palette stuffing," JCTVC-R0348, June 2014.

[15] P. Lai, S. Liu, T.-D. Chuang, Y.-W. Huang, and S. Lei, "Non-RCE4: Major color table (palette) sharing," JCTVC-P0153, Jan. 2014.

[16] 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," JCTVC-R0057, June 2014.

[17] Y.-C. Sun, T.-D. Chuang, Y.-W. Chen, Y.-W. Huang, and S. Lei, "Non-RCE4: Removal of syntax redundancy in RCE4 Test2," JCTVC-P0095, Jan. 2014.

[18] https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.2+SCM-3.0/

[19] H. Yu, R. Cohen, K. Rapaka, J. Xu, "Common test conditions for screen content coding," JCTVC-S1015, Oct. 2014.

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