

# **AHG17/Non-CE8: APS support for Initial Palette Predictor Entry (JVET-P0577)**

**Hyeongmun Jang**

# Summary

---

- Motivation,
  - In HEVC SCC extension, It was observed that in the presence of tiles and wave-front parallel processing palette coding resulted in substantial coding efficiency loss.
  - It was asserted that it took a time to rebuild the palette predictor entry list.
  - For SCC, the content provider or “smart” encoder could generate palette entry and signal entry information to help compression performance by initializing the base palette on the transmitted palette entry.
  - In HEVC SCC extension, the predictor palette initializer is transmitted in SPS.
  - Amount of the palette entry initializer could be about 236Byte.  
(10-bit \* 3-component \* 63-predictor in palette)
  - The Adaptation Parameter Set has been identified a more practical choice for the palette initializer rather than SPS, PPS or Slice header

# Proposed method.

- Signal the initial palette predictor data in APS.

adaptation_parameter_set_rbsp( ) {	<b>Descriptor</b>
<b>adaptation_parameter_set_id</b>	u(5)
<b>aps_params_type</b>	u(3)
if( aps_params_type == ALF_APS )	
alf_data( )	
else if( aps_params_type == LMCS_APS )	
lmcs_data( )	
else if( aps_params_type == SCALING_APS )	
scaling_list_data( )	
else if( aps_params_type == PLT_APS )	
palette_entry_data( )	
<b>aps_extension_flag</b>	u(1)
if( aps_extension_flag )	
while( more_rbsp_data( ) )	
<b>aps_extension_data_flag</b>	u(1)
rbsp_trailing_bits( )	
}	

aps_params_type	Name of aps_params_type	Type of APS parameters
0	ALF_APS	ALF parameters
1	LMCS_APS	LMCS parameters
2	SCALING_APS	Scaling list parameters
3	PLT_APS	Palette entry parameters
4..7	Reserved	Reserved

# Proposed method.

- Syntax for the palette predictor initializer in APS.

palette_entry_data( ) {	Descriptor
palette_entry_bit_depth_luma_minus8	ue(v)
palette_entry_bit_depth_chroma_minus8	ue(v)
num_signalled_palette_entry_minus1	ue(v)
for( cIdx = 0; cIdx < 3; cIdx++ )	
for( i = 0; i <= num_signalled_palette_entries_minus1; i++ )	
initial_predictor_palette_entries[ cIdx ][ i ]	ue(v)
}	

**palette\_entry\_bit\_depth\_luma\_minus8** specifies the bit depth of the samples of the palette entry  $PltBitDepth$  as follows:

$$PltBitDepth_Y = 8 + \text{palette\_entry\_bit\_depth\_luma\_minus8}$$

**palette\_entry\_bit\_depth\_luma\_minus8** shall be in the range of 0 to 8, inclusive.

**palette\_entry\_bit\_depth\_chroma\_minus8** specifies the bit depth of the samples of the palette entry  $PltBitDepth$  as follows:

$$PltBitDepth_C = 8 + \text{palette\_entry\_bit\_depth\_chroma\_minus8}$$

**palette\_entry\_bit\_depth\_chroma\_minus8** shall be in the range of 0 to 8, inclusive.

**num\_signalled\_palette\_entry\_minus1** specifies the number of entries in the APS palette predictor entry.

**initial\_predictor\_palette\_entries[ cIdx ][ i ]** specifies the value of the  $cIdx$ -th component of the  $i$ -th palette entry in the APS that is used to initialize the array `PredictorPaletteEntries`. For values of  $i$  in the range of 0 to  $\text{num\_signalled\_palette\_entry\_minus1}$ , inclusive, the value of `initial_predictor_palette_entries[ 0 ][ i ]` shall be in the range of 0 to  $(1 \ll PltBitDepth_Y) - 1$ , inclusive, and the values of `initial_predictor_palette_entries[ 1 ][ i ]` and `initial_predictor_palette_entries[ 2 ][ i ]` shall be in the range of 0 to  $(1 \ll PltBitDepth_C) - 1$ , inclusive.

# Proposed method.

- Slice header syntax for indicating .

slice_header( ) {	Descriptor
<b>slice_pic_parameter_set_id</b>	ue(v)
...	
if( sps_palette_enabled_flag ) {	
<b>slice_palette_enabled_flag</b>	u(1)
<b>slice_palette_entries_present_flag</b>	u(1)
if( slice_palette_entries_present_flag )	
<b>slice_palette_entries_aps_id</b>	u(2)
}	
...	
}	

**slice\_palette\_enabled\_flag** equal to 1 specifies that palette prediction mode is enabled for the current slice. **slice\_palette\_enabled\_flag** equal to 0 specifies that palette prediction mode is not enabled for the current slice. When **slice\_palette\_enabled\_flag** equal to 1 specifies that palette prediction mode is enabled for the current slice.

**slice\_palette\_entries\_present\_flag** equal to 1 specifies that the palette predictor used for the current slice is derived based on the palette entries data contained in the referenced palette entries APS and PredictorPaletteSize[cIdx] is set equal to num\_signaled\_palette\_entry\_minus1 + 1, cIdx=0..2. The **slice\_palette\_entries\_present\_flag** equal to 0 specifies that the palette predictor used for the current picture is reset to 0 and PredictorPaletteSize[cIdx] is set equal to 0, cIdx=0..2.. When not present, the value of **slice\_palette\_entries\_present\_flag** is inferred to be equal to 0.

**slice\_palette\_aps\_id** specifies the adaptation\_parameter\_set\_id of the PLT\_APS that the slice refers to. The TemporalId of the APS NAL unit having aps\_params\_type equal to PLT\_APS and adaptation\_parameter\_set\_id equal to **slice\_palette\_aps\_id** shall be less than or equal to the TemporalId of the coded slice NAL unit.

When present, the value of **slice\_palette\_aps\_id** shall be the same for all slices of a picture.

# Conclusion

---

- It is recommended to adopt in Next VTM.