JCTVC-T0064 working draft changes

On top of:

R. Joshi and J. Xu, “HEVC screen content coding draft text 2,” in *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 19th Meeting*, JCTVC-S1005, Strasbourg, FR, 17–24 Oct. 2014.

#### 7.3.8.8 Palette syntax

|  |  |
| --- | --- |
| palette\_coding( x0, y0, nCbS ) { | Descriptor |
| **~~palette\_share\_flag~~**~~[ x0 ][ y0 ]~~ | ~~ae(v)~~ |
| ~~if( !palette\_share\_flag[ x0 ][ y0 ] ) {~~ |  |
| palettePredictionFinished = 0 |  |
| paletteNumPredictedEntries = 0 |  |
| for( i = 0; i < PredictorPaletteSize  && !palettePredictionFinished &&  paletteNumPredictedEntries < palette\_max\_size; i++ ) { |  |
| **palette\_predictor\_run** | ae(v) |
| if( palette\_predictor\_run != 1 ) |  |
| if( palette\_predictor\_run > 1 ) |  |
| i += palette\_predictor\_run − 1 |  |
| PalettePredictorEntryReuseFlag[ i ] = 1 |  |
| paletteNumPredictedEntries ++ |  |
| } else |  |
| palettePredictionFinished = 1 |  |
| } |  |
| if( paletteNumPredictedEntries < palette\_max\_size ) |  |
| **palette\_num\_signalled\_entries** | ae(v) |
| for( cIdx = 0; cIdx < 3; cIdx++ ) |  |
| for( i = 0; i < palette\_num\_signalled\_ entries; i++ ) |  |
| **palette\_entry** | ae(v) |
| ~~}~~ |  |
| … |  |

#### Table 9‑43 – Assignment of ctxInc to syntax elements with context coded bins

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| … | … | … | … | … | … | … |
| palette\_num\_signalled\_entries | ~~Bypass~~ 0 | bypass | bypass | bypass | bypass | bypass |
| … | … | … | … | … | … | … |
| ~~palette\_share\_flag~~ | ~~0~~ | ~~na~~ | ~~na~~ | ~~na~~ | ~~na~~ | ~~na~~ |
| … |  |  |  |  |  |  |

#### Table 9-38 – Syntax elements and associated binarizations

|  |  |  |  |
| --- | --- | --- | --- |
| palette\_coding( ) | ~~palette\_share\_flag~~ | ~~FL~~ | ~~cMax = 1~~ |

**~~palette\_share\_flag~~**~~[ x0 ][ y0 ] equal to 1 specifies that the palette for the current coding unit is derived by copying the first PreviousPaletteSize entries from the predictor palette. The variable PreviousPaletteSize is derived as specified in subclause 8.4.5.2.8. palette\_share\_flag [ x0 ][ y0 ] equal to 0 specifies the palette for the current coding unit is specified as a combination of palette entries from previous coding units and new palette entries which are explicitly signalled.~~

~~The variable PalettePredictorEntryReuseFlag[ i ]~~~~equal to 1~~~~specifies that the i-th entry in the predictor palette is reused in the current palette. PalettePredictorEntryReuseFlag[ i ] equal to 0~~~~specifies that the i-th entry in the predictor palette is an entry in the current palette. All elements of the array PalettePredictorEntryReuseFlag[ i ] are initialized to be equal to zero.~~

~~If palette\_share\_flag [ x0 ][ y0 ] is equal to 1, the array PalettePredictorEntryReuseFlag[ i ] is further modified as follows:~~

~~for( i = 0; i < PreviousPaletteSize; i++ ) (7‑70)  
 PalettePredictorEntryReuseFlag[ i ] = 1~~

**palette\_num\_signalled\_entries** specifies the number of entries in the current palette that are explicitly signalled.

When palette\_num\_signalled\_ entries is not present, it is inferred to be equal to 0.

The variable CurrentPaletteSize specifies the size of the current palette and is derived as follows:

* ~~If palette\_share\_flag [ x0 ][ y0 ] is equal to 1,~~

~~CurrentPaletteSize = PreviousPaletteSize (7‑71)~~

* ~~Otherwise (palette\_share\_flag [ x0 ][ y0 ] is equal to 0)~~

CurrentPaletteSize = paletteNumPredictedEntries + palette\_num\_signalled\_entries (7‑72)

The value of CurrentPaletteSize shall be in the range of 0 to palette\_max\_size, inclusive.

**palette\_entry** specifiesthe value of a component in a palette entry for the current palette.

The variable PredictorPaletteEntries[ cIdx ][ i ]specifiesthe i-th element in the predictor palette for the colour component cIdx.

The variable CurrentPaletteEntries[ cIdx ][ i ]specifiesthe i-th element in the current palette for the colour component cIdx and is derived as follows:

~~When palette\_share\_flag [ x0 ][ y0 ] is equal to 1, CurrentPaletteEntries is derived as follows:~~

* ~~If palette\_share\_flag [ x0 ][ y0 ] is equal to 1,~~

~~for( cIdx = 0; cIdx < 3; cIdx++ )  
 for( i = 0; i < PreviousPaletteSize; i++ ) (7‑73)  
 CurrentPaletteEntries[ cIdx ][ i ] = PredictorPaletteEntries[ cIdx ][ i ]~~

* ~~Otherwise (palette\_share\_flag [ x0 ][ y0 ] is equal to 0)~~

numPredictedPaletteEntries = 0  
for( i = 0; i < PredictorPaletteSize; i++ )   
 if( PalettePredictorEntryReuseFlag[ i ] **=** **=** 1 ) {  
 for( cIdx = 0; cIdx < 3; cIdx++ )   
 CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] = PredictorPaletteEntries[ cIdx ][ i ]  
 numPredictedPaletteEntries++   
 }  
for( cIdx = 0; cIdx < 3; cIdx++ ) (7‑74)  
 for( i = 0; i < num\_signalled\_palette\_entries; i++ )   
 CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] = palette\_entry

Table 9‑4 – Association of ctxIdx and syntax elements for each initializationType in the initialization process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| … | … | … | … | … | … |
| palette\_coding( ) | palette\_run\_msb\_id\_plus1 | Table 9‑40 | 0..7 | 8..15 | 16..23 |
| cu\_qp\_delta\_palette\_abs | Table 9‑24 | 0..1 | 2..3 | 4..5 |
| cu\_chroma\_qp\_palette\_offset\_flag | Table 9‑34 | 0 | 1 | 2 |
| cu\_chroma\_qp\_palette\_offset\_idx | Table 9‑35 | 0 | 1 | 2 |
| palette\_num\_signalled\_entries | Table 9-xx | 0 | 1 | 2 |
| … | … | … | … | … | … |

Table 9‑xx – Values of initValue for ctxIdx of palette\_num\_signalled\_entries

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **ctxIdx of palette\_num\_signalled\_entries** | | |
| **0** | **1** | **2** |
| **initValue** | 154 | 154 | 154 |