AhG way of support palette mo­de for non 4:4:4 chroma subsampling working draft

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.

#### 7.3.8.5 Coding unit syntax

|  |  |
| --- | --- |
| if( palette\_mode\_enabled\_flag && ~~ChromaArrayType = = 3 &&~~   CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA && !intra\_bc\_flag[ x0 ][ y0 ] ) |  |
| **palette\_mode\_flag**[ x0 ][ y0 ] | ae(v) |
| if( palette\_mode\_flag[ x0 ][ y0 ] ) { |  |
| if( ChromaArrayType = = 0 | | ChromaArrayType = = 3 ) |  |
| palette\_coding( x0, y0, nCbS, ChromaArrayType ) |  |
| else { |  |
| palette\_coding( x0, y0, nCbS, 1 ) |  |
| palette\_coding( x0, y0, nCbS, 2 ) |  |
| } |  |
| } else { |  |
| … |  |

#### 7.3.8.8 Palette syntax

|  |  |
| --- | --- |
| palette\_coding( x0, y0, nCbS, nComp ) { | Descriptor |
| compBegin = ( nComp = = 2 ) ? 1 : 0 |  |
| **palette\_share\_flag**[ x0 ][ y0 ] | ae(v) |
| if( !palette\_share\_flag[ x0 ][ y0 ] ) { |  |
| palettePredictionFinished = 0 |  |
| paletteNumPredictedEntries = 0 |  |
| for( i = 0; i < PredictorPaletteSize[ compBegin ]  && !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[ compBegin ][ i ] = 1 |  |
| paletteNumPredictedEntries ++ |  |
| } else |  |
| palettePredictionFinished = 1 |  |
| } |  |
| if( paletteNumPredictedEntries < palette\_max\_size ) |  |
| **palette\_num\_signalled\_entries** | ae(v) |
| for( cIdx = compBegin; cIdx < compBegin + nComp ; cIdx++ ) |  |
| for( i = 0; i < palette\_num\_signalled\_ entries; i++ ) |  |
| **palette\_entry** | ae(v) |
| } |  |
| if( currentPaletteSize != 0 ) |  |
| **palette\_escape\_val\_present\_flag** | ae(v) |
| if( palette\_escape\_val\_present\_flag ) { |  |
| if( !compBegin && cu\_qp\_delta\_enabled\_flag && !IsCuQpDeltaCoded ) { |  |
| **cu\_qp\_delta\_palette\_abs** | ae(v) |
| if( cu\_qp\_delta\_palette\_abs ) |  |
| **cu\_qp\_delta\_palette\_sign\_flag** | ae(v) |
| } |  |
| if( nComp > 1  && cu\_chroma\_qp\_offset\_enabled\_flag && !IsCuChromaQpOffsetCoded ) { |  |
| **cu\_chroma\_qp\_palette\_offset\_flag** | ae(v) |
| if( cu\_chroma\_qp\_offset\_flag && chroma\_qp\_offset\_list\_len\_minus1 > 0 ) |  |
| **cu\_chroma\_qp\_palette\_offset\_idx** | ae(v) |
| } |  |
| } |  |
| if( indexMax > 0) |  |
| **palette\_transpose\_flag** | ae(v) |
| scanPos = 0 |  |
| while( scanPos < nCbS \* nCbS ) { |  |
| xC = x0 + travScan[ scanPos ][ 0 ] |  |
| yC = y0 + travScan[ scanPos ][ 1 ] |  |
| if( scanPos > 0) { |  |
| xcPrev = x0 + travScan[ scanPos − 1 ][ 0 ] |  |
| ycPrev = y0 + travScan[ scanPos − 1 ][ 1 ] |  |
| } |  |
| if( indexMax > 0 && scanPos > = nCbS && palette\_run\_type\_flag[ xcPrev ][ ycPrev ]  ! = COPY\_ABOVE\_MODE ) { |  |
| **palette\_run\_type\_flag**[ xC ][ yC ] | ae(v) |
| } |  |
| if( palette\_run\_type\_flag[ xC ][ yC ] = = COPY\_INDEX\_MODE &&   adjustedIndexMax > 0) |  |
| **palette\_index\_idc** | ae(v) |
| if( indexMax > 0 ) { |  |
| maxPaletteRun = nCbS \* nCbS – scanPos – 1 |  |
| if( maxPaletteRun > 0 ) { |  |
| **palette\_run\_msb\_id\_plus1** | ae(v) |
| if( palette\_run\_msb\_id\_plus1 > 1 ) |  |
| **palette\_run\_refinement\_bits** | ae(v) |
| } |  |
| } else |  |
| paletteRun = nCbS \* nCbS – 1 |  |
| runPos = 0 |  |
| while ( runPos < = paletteRun ) { |  |
| xR = x0 + travScan[ scanPos ][ 0 ] |  |
| yR = y0 + travScan[ scanPos ][ 1 ] |  |
| if( palette\_run\_type\_flag[ xC ][ yC ] = = COPY\_INDEX\_MODE &&   paletteIndex = = indexMax ) { |  |
| PaletteSampleMode[ compBegin ][ xR ][ yR ] = ESCAPE\_MODE |  |
| PaletteIndexMap[ compBegin ][ xR ][ yR ] = paletteIndex |  |
| for( cIdx = 0; cIdx < 3; cIdx++ ) { |  |
| **palette\_escape\_val** | ae(v) |
| PaletteEscapeVal[ cIdx ][ xR ][ yR ] = palette\_escape\_val |  |
| } |  |
| } else if( palette\_run\_type\_flag[ xC ][ yC ] = = COPY\_INDEX\_MODE ) { |  |
| PaletteSampleMode[ compBegin ][ xR ][ yR ] = COPY\_INDEX\_MODE |  |
| PaletteIndexMap[ compBegin ][ xR ][ yR ] = paletteIndex |  |
| } else { |  |
| PaletteSampleMode[ compBegin ][ xR ][ yR ] = COPY\_ABOVE\_MODE |  |
| PaletteIndexMap[ compBegin ][ xR ][ yR ] = PaletteIndexMap[ compBegin ] [ xR ][ yR − 1 ] |  |
| } |  |
| runPos++ |  |
| scanPos++ |  |
| } |  |
| } |  |
| } |  |