|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  9th Meeting: Sapporo, JP, 3 – 9 July 2014 | Document: JCT3V-J0050 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **IC improvement for chroma** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Junghak Nam  Sehoon Yea  19, Yangjae-daero 11gil, Seocho-gu Seoul 137-130, Korea | Tel: Email: | +82-2-6912-6477 [junghak.nam@lge.com](mailto:junghak.nam@lge.com) |
| *Source:* | LG Electronics | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The newly added parts compared to 3D-HEVC working draft 3 are highlighted in green and the removed parts are marked with ~~strikethrough~~.

Method 2: Offset model of chroma IC

I.8.5.5.3.3.6.1 Derivation process for illumination compensation mode availability and parameters

1. For X being replaced by 0 and 1, when puIcFlagLX is equal to 1, the variables icWeightLX, and icOffsetLX are derived by the following ordered steps:
2. 1. ……
3. 2. ……

3. The derivation process for illumination compensation parameters as specified in subclause is invoked, with the list of neighbouring samples in the current picture curNeighSampleList, the list of neighbouring samples in the reference picture refNeighSample, a variable cIdx specifying colour component index and the number of neighbouring samples numNeighSamlesLX as inputs and the illumination parameters icWeightLX, and icOffsetLX as outputs.

I.8.5.3.3.6.2 Derivation process for illumination compensation parameters

Inputs to this process are:

* a list curSampleList specifying the current samples,
* a list refSampleList specifying the reference samples,
* a variable numSamples specifying the number of elements in curSampleList and refSampleList,
* a bit depth of samples, bitDepth,
* a variable cIdx specifying colour component index.

Outputs of this process are:

* a variable icWeight specifying a weight for illumination compensation,
* a variable icOffset specifying a offset for illumination compensation.

The variable precShift is set equal to Max( 0, bitDepth − 12 ).

The variables sumRef, sumCur, sumRefSquare and sumProdRefCur are set equal to 0 and the following applies for i ranging from 0 to numSamples / 2 − 1, inclusive:

sumRef += refSampleList[ 2 \* i ] (‑208)

sumCur += curSampleList[ 2 \* i ] (‑209)

The variable avgShift and avgOffset are derived as follows:

avgShift = Log2( numSamples / 2) (‑212)

avgOffset = 1 << ( avgShift − 1 ) (‑213)

* If cIdx is equal to 0, the following applies:

sumRefSquare += ( refSampleList[ 2 \* i ] \* refSampleList[ 2 \* i ] ) >> precShift (‑210)

sumProdRefCur += ( refSampleList[ 2 \* i ] \* curSampleList[ 2 \* i ] ) >> precShift (‑211)

~~The variable avgShift and avgOffset are derived as follows:~~

~~avgShift = Log2( numSamples / 2) (‑212)~~

~~avgOffset = 1 << ( avgShift − 1 ) (‑213)~~

The variables numerDiv and denomDiv are derived as follows:

denomDiv= ( ( sumRefSquare + ( sumRefSquare >> 7 ) )  <<  avgShift )  
  − ( sumRef \* sumRef ) >> precShift (‑214)

numerDiv= Clip3( 0, 2 \* denomDiv, ( ( sumProdRefCur + ( sumRefSquare >> 7 ) )  <<  avgShift )  
  − ( sumRef \* sumCur ) >> precShift ) (‑215)

The variables shiftNumer and shiftDenom are derived as follows:

shiftDenom = Max( 0, Floor( Log2( Abs( denomDiv ) ) ) − 5 ) (‑216)

shiftNumer = Max( 0, shiftDenom − 12 ) (‑217)

The variables sNumerDiv and sDenomDiv are derived as follows:

sDenomDiv = denomDiv >> shiftDenom (‑218)

sNumerDiv = numerDiv >> shiftNumer (‑219)

The value of variable divCoeff is derived from depending on sDenomDiv and the variables icWeight, and icOffset are derived as follows:

icWeight = ( sNumerDiv \* divCoeff ) >> ( shiftDenom − shiftNumer + 10 ) (‑220)

icOffset = ( sumCur − ( ( icWeight \* sumRef ) >> 5 ) + avgOffset ) >> avgShift (‑221)

* Otherwise (cIdx is equal to 1 or 2), the following applies:

icWeight = 32 (‑2xx)

icOffset = ( sumCur − sumRef  + avgOffset ) >> avgShift (‑2xx)