|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  5th Meeting: Vienna, AT, 27 July – 2 Aug. 2013 | Document: JCT3V-E0046 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE4-related: Resampling in IC parameter derivation and 4x4 Chroma removal** | | |
| *Status:* | Input Document to JCT-3V | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Tomohiro Ikai  1-9-2 Nakase, Mihama-ku, Chiba-shi, Chiba 261-8520  JAPAN | Tel: Email: | +81-43-299-8526 [ikai.tomohiro@sharp.co.jp](mailto:ikai.tomohiro@sharp.co.jp) |
| *Source:* | Sharp Corporation | | |

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

# Abstract

This contribution presents a further simplification on top of CE4 test 1.1. In addition to regression improvement and several simplifications by CE4 test 1.1, it additionally simplifies illumination compensation process by resampling reference pixels in IC parameter derivation and removal of 4x4 chroma block. The experimental result reportedly shows 0.2 % gain in video, total video and synthesis respectively.

# Proposal

Additional simplification on top of CE4 test1.1 is proposed as follows.

* chroma 4x4 illumination compensation is disabled
* illumination compensation parameters is derived based on 2:1 sub-sampling pixels.

# Proposed text

Test 1.1 part is highlighted with Green(added part) and Gray(removed part).

Additional simplification by the proposal is highlighted with Yellow.

**H.8.5.2.2 Decoding process for inter prediction samples**

* 1. Depending on ic\_flag and nPbW, the arrays predSamplesL ,predSampleCb, and predSampleCr are derived as specified in the following:
     + If ic\_flag is equal to 0, the following applies.
       - The array predSampleL of the prediction samples of luma component is derived by invoking the weighted sample prediction process specified in subclause 8.5.2.2.3 with the luma location ( xB, yB ), the width and the height of the current luma prediction block nPbW, nPbH, and the sample arrays predSamplesL0L and predSamplesL1L as well as predFlagL0, predFlagL1, refIdxL0, refIdxL1 and cIdx equal to 0 given as input.
     + Otherwise ( ic\_flag is equal to 1), the following applies.
       - The array predSampleL of the prediction samples of luma component is derived by invoking the illumination compensated sample prediction process specified in subclause , with the luma location ( xC, yC ), the size of the current luma coding block nCS, the luma location ( xB, yB ), the width and the height of the current luma prediction block nPbW, nPbH, and the sample arrays predSamplesL0L and predSamplesL1L as well as predFlagL0, predFlagL1, refIdxL0, refIdxL1, mvL0, mvL1 and cIdx equal to 0 given as input.
     + If ic\_flag is equal to 0 or nPbW is less than 8, the following applies.
       - The array predSampleCb of the prediction samples of component Cb is derived by invoking the weighted sample prediction process specified in subclause 8.5.2.2.3 with the chroma location ( xB/2, yB/2 ), the width and the height of the current chroma prediction block nPbWCb set equal to nPbW/2, nPbHCb set equal to nPbH/2, and the sample arrays predSamplesL0Cb and predSamplesL1Cb as well as predFlagL0, predFlagL1, refIdxL0, refIdxL1, and cIdx equal to 1 given as input.
       - The array predSampleCr of the prediction samples of component Cr is derived by invoking the weighted sample prediction process specified in subclause 8.5.2.2.3 with the chroma location ( xB/2, yB/2 ), the width and the height of the current chroma prediction block nPbWCr set equal to nPbW/2, nPbHCr set equal to nPbH/2, and the sample arrays predSamplesL0Cr and predSamplesL1Cr as well as predFlagL0, predFlagL1, refIdxL0, refIdxL1, and cIdx equal to 2 given as input.
     + Otherwise (ic\_flag is equal to 1 and nPbW is equal to or more than 8), the following applies.
       - The array predSampleCb of the prediction samples of component Cb is derived by invoking the illumination compensated sample prediction process specified in subclause , with the luma location ( xC, yC ), the size of the current luma coding block nCS, with the chroma location ( xB/2, yB/2 ), the width and the height of the current chroma prediction block nPbWCb set equal to nPbW/2, nPbHCb set equal to nPbH/2, and the sample arrays predSamplesL0Cb and predSamplesL1Cb as well as predFlagL0, predFlagL1, refIdxL0, refIdxL1, mvCL0, mvCL1, and cIdx equal to 1 given as input.
       - The array predSampleCr of the prediction samples of component Cr is derived by invoking the illumination compensated sample prediction process specified in subclause , with the luma location ( xC, yC ), the size of the current luma coding block nCS, with the chroma location ( xB/2, yB/2 ), the width and the height of the current chroma prediction block nPbWCr set equal to nPbW/2, nPbHCr set equal to nPbH/2, and the sample arrays predSamplesL0Cr and predSamplesL1Cr as well as predFlagL0, predFlagL1, refIdxL0, refIdxL1, mvCL0, mvCL1, and cIdx equal to 2 given as input.

**H.8.5.2.2.5 Illumination compensated sample prediction process**

…

Depending on the value of predFlagL0 and predFlagL1, the prediction samples predSamples[ x ][ y ] with x = 0..(nPbW)−1 and y = 0..(nPbH)−1 are derived as follows.

* If predFlagL0 is equal to 1 and predFlagL1 is equal to 0,
  + 1. clipPredVal= Clip3( 0, ( 1 << bitDepth ) − 1, ( predSamplesL0[ x ][ y ] + offset1 ) >> shift1 ) (H‑148)
    2. predSamples[ x ][ y ] = !puIcFlagL0 ? clipPredVal :   
       ( Clip3( 0, ( 1 << bitDepth ) − 1, ( clipPredVal \* icWeightL0 ) >> 5~~icShiftL0~~ ) + icOffsetL0 ) (H‑149)
* Otherwise, if predFlagL0 is equal to 0 and predFlagL1 is equal to 1,
  + 1. clipPredVal = Clip3( 0, ( 1 << bitDepth ) − 1, ( predSamplesL1[ x ][ y ] + offset1 ) >> shift1 ) (H‑150)
    2. predSamples[ x ][ y ] = !puIcFlagL1 ? clipPredVal :   
       ( Clip3( 0, ( 1 << bitDepth ) − 1, ( clipPredVal \* icWeightL1 ) >> 5~~icShiftL1~~ ) + icOffsetL1 ) (H‑151)
* Otherwise,
  + 1. predVal0 = !puIcFlagL0 ? predSamplesL0[ x ][ y ] :   
       ( ( predSamplesL0[ x ][ y ] \* icWeightL0 ) >> 5~~icShiftL0~~  ) + ( icOffsetL0 << shift1 ) ) (H‑152)
    2. predVal1 = !puIcFlagL1 ? predSamplesL1[ x ][ y ] :   
       ( ( predSamplesL1[ x ][ y ] \* icWeightL1 ) >> 5~~icShiftL1~~ ) + ( icOffsetL1 << shift1 ) ) (H‑153)
    3. predSamples[ x ][ y ] =   
       Clip3( 0, ( 1 << bitDepth ) − 1, ( predVal0 + predVal1 + offset2 ) >> shift2 ) (H‑154)

**H.8.5.2.2.5.2** **Derivation process for illumination compensation parameters**

…

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.

* 1. sumRef += refSampleList[ 2\*i ] (H‑170)
  2. sumCur += curSampleList[ 2\*i ] (H‑171)
  3. sumRefSquare += ( refSampleList[ 2\*i ] \* refSampleList[ 2\*i ] ) >> precShift (H‑172)
  4. sumProdRefCur += ( refSampleList[ 2\*i ] \* curSampleList[ 2\*i ] ) >> precShift (H‑173)

…

The variable avgShift and avgOffset specifying the bit shift and offset needed for averaging are derived as

* 1. avgShift = Ceil( Log2( numSamples/2 ) ) − precShift (H‑180)
  2. avgOffset = 1 << ( avgShift − 1 ) (H‑181)

avgShift = Ceil( Log2( numSamples/2 ) ) − precShift (H‑180)

…

The variables numerDiv and denomDiv specifying numerator and denominator of a following divisions are derived as.

numerDiv= ( (sumProdRefCur + (sumRefSquare>>7)) << avgShift ) – sumRef \* sumCur (H‑182)

denomDiv= ( (sumRefSquare + (sumRefSquare>>7)) << avgShift ) – sumRef \* sumRef (H‑183)

numerDiv= Clip3(0, 2\* denomDiv, numerDiv)

The variables psShiftNumer and psShiftDenom are derived as

psShiftDenom = Max( 0, Floor( Log2( Abs( denomDiv ) ) ) − 5) (H‑185)

~~psShiftNumer = Max( 0, Floor( Log2( Abs( numerDiv ) ) ) − 14) (‑184)~~

psShiftNumer = Max( 0, psShiftDenom − 12) (‑184)

The variables psNumerDiv and psDenomDiv are derived as

* 1. psNumerDiv = numerDiv >> psShiftNumer (H‑186)
  2. psDenomDiv = denomDiv >> psShiftDenom (H‑187)

The variable psIcWeight specifying the shifted weight for illumination compensation is derived as specified in the following.

* + ~~If psDenomDiv is greater than 0, the following applies,~~
    - The value of variable divCoeff is derived from Table H‑11 depending on psDenomDiv.
    - The value of psIcWeight is derived as
      * 1. psIcWeight = psNumerDiv \* divCoeff (H‑188)
  + ~~Otherwise( psDenomDiv is less or equal to 0), psIcWeight is set equal to 0.~~

~~The variable icShift specifying a bit shift for illumination compensation is set equal to 13.~~

The variable invPsShift is derived as

* 1. invPsShift = psShiftDenom – psShiftNumer + 15 – 5 (H‑189)

The variable icWeight is derived as specified in the following

* + ~~If invPsShift is less than 0, the following applies:~~
    - 1. ~~invPsIcWeight = Clip3( psIcWeight << ( Abs( invPsShift ) ), −2~~~~15~~~~, 2~~~~15~~ ~~− 1~~~~) (‑190)~~
  + ~~Otherwise, ( invPsIcWeight is greater than or equal to 0), the following applies:~~ 
    - 1. icWeight = psIcWeight >> invPsShift (H‑191)

~~The variable icWeight specifying a weight for illumination compensation with 7 bit precision is derived as specified in the following:~~

* + ~~If invPsIcWeight is greater than or equal to −2~~~~6~~ ~~and less than 2~~~~6~~~~, the following applies.~~
    - 1. ~~icWeight = invPsIcWeight (‑192)~~
  + ~~Otherwise, ( invPsIcWeight is less than −2~~~~6~~ ~~or greater than or equal to 2~~~~6~~ ~~), the following applies.~~
    - 1. ~~decIcShift = Max( 0, Floor(Log2( Abs( icWeight ) ) − 5 ) ) (‑193)~~
      2. ~~[Ed (GT): In software a function counting leading zero ones is utilized to derive decIcShift. Does this match with draft text?]~~
      3. ~~icWeight = invPsIcWeight >> decIcShift (‑194)~~
      4. ~~icShift −= decIcShift~~ (‑195)

The variable icOffset specifying an offset for illumination compensation is derived as:

icOffset = ( sumCur – ( ( icWeight\*sumRef ) >>  5 ) + avgOffset ) >>  avgShift (H‑196)

# Simulation results

Table 1 shows the simulation results. Table 2 shows the comparison between proposal and CE4 test 1.1. Table 2 shows the additional proposal doesn’t decrease coding efficiency.

Table 1 Resuls of CE4 test1.1 + additional simplification results

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | video 0 | video 1 | video 2 | video PSNR / video bitrate | video PSNR / total bitrate | synth PSNR / total bitrate | enc time | dec time | ren time |
| Balloons | 0.0% | -0.9% | -0.8% | -0.3% | -0.3% | -0.3% | 100.5% | 100.1% | 99.6% |
| Kendo | 0.0% | -0.6% | -1.3% | -0.4% | -0.3% | -0.2% | 99.4% | 95.3% | 97.3% |
| Newspaper\_CC | 0.0% | 0.0% | -0.7% | -0.1% | -0.1% | -0.4% | 100.6% | 101.6% | 100.1% |
| GT\_Fly | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.5% | 96.7% | 97.9% |
| Poznan\_Hall2 | 0.0% | -0.9% | -2.2% | -0.7% | -0.6% | -0.4% | 100.2% | 99.8% | 99.6% |
| Poznan\_Street | 0.0% | -0.2% | -0.6% | -0.1% | -0.1% | -0.1% | 98.7% | 97.9% | 97.6% |
| Undo\_Dancer | 0.0% | 0.1% | -0.1% | 0.0% | 0.0% | 0.1% | 98.5% | 98.5% | 98.2% |
| 1024x768 | 0.0% | -0.5% | -0.9% | -0.3% | -0.2% | -0.3% | 100.2% | 99.0% | 99.0% |
| 1920x1088 | 0.0% | -0.3% | -0.7% | -0.2% | -0.2% | -0.1% | 99.2% | 98.2% | 98.3% |
| **average** | **0.0%** | **-0.4%** | **-0.8%** | **-0.2%** | **-0.2%** | **-0.2%** | **99.6%** | **98.6%** | **98.6%** |

Table 2 Resuls of proposal (anchor CE4 test1.1)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | video 0 | video 1 | video 2 | video PSNR / video bitrate | video PSNR / total bitrate | synth PSNR / total bitrate | enc time | dec time | ren time |
| Balloons | 0.0% | 0.1% | 0.1% | 0.05% | 0.03% | -0.10% | 100.0% | 97.3% | 98.5% |
| Kendo | 0.0% | 0.1% | 0.4% | 0.10% | 0.11% | 0.10% | 99.0% | 95.4% | 96.4% |
| Newspaper\_CC | 0.0% | 0.1% | 0.1% | 0.03% | 0.00% | 0.02% | 99.7% | 100.2% | 99.3% |
| GT\_Fly | 0.0% | 0.0% | 0.0% | 0.00% | 0.00% | -0.02% | 98.9% | 97.7% | 98.0% |
| Poznan\_Hall2 | 0.0% | 0.1% | -0.5% | -0.07% | -0.05% | -0.12% | 99.3% | 99.9% | 99.7% |
| Poznan\_Street | 0.0% | -0.2% | 0.0% | -0.03% | -0.02% | -0.04% | 98.3% | 97.9% | 97.5% |
| Undo\_Dancer | 0.0% | 0.1% | 0.0% | 0.01% | 0.01% | 0.04% | 97.2% | 98.1% | 98.4% |
| 1024x768 | 0.0% | 0.1% | 0.2% | 0.06% | 0.05% | 0.01% | 99.6% | 97.6% | 98.1% |
| 1920x1088 | 0.0% | 0.0% | -0.1% | -0.02% | -0.02% | -0.03% | 98.4% | 98.4% | 98.4% |
| **average** | **0.0%** | **0.0%** | **0.0%** | **0.01%** | **0.01%** | **-0.02%** | **98.9%** | **98.1%** | **98.3%** |

# Complexity assessment

Table 3 show the complexity assessment results of HTM70, CE test (test1.1 and test1.3) and proposal, respectively. The data (number of operations) is calculated based on CE4 complexity assessment template.

Table 3. Comparison of Number of operations (per 8x8 CU)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HTM70 | Summary | Comparison | Add/Sub | Mul | ShiftC | ShiftA | Clip | Table | GetMSB |
|  | 8x8CU | 36 | 266 | 172 | 0 | 123 | 99 | 3 | 9 |
|  | 16x16CU | 9 | 171 | 131 | 0 | 103 | 97 | 1 | 2 |
|  | 32x32CU | 2 | 131 | 113 | 0 | 98 | 96 | 0 | 1 |
|  | 64x64CU | 1 | 113 | 104 | 0 | 96 | 96 | 0 | 0 |
| Test1.1 | Summary | Comparison | Add/Sub | Mul | ShiftC | ShiftA | Clip | Table | GetMSB |
|  | 8x8CU | 24 | 257 | 175 | 105 | 21 | 99 | 3 | 3 |
|  | 16x16CU | 6 | 168 | 132 | 98 | 5 | 97 | 1 | 1 |
|  | 32x32CU | 2 | 130 | 113 | 97 | 1 | 96 | 0 | 0 |
|  | 64x64CU | 0 | 113 | 104 | 96 | 0 | 96 | 0 | 0 |
| Test1.3 | Summary | Comparison | Add/Sub | Mul | ShiftC | ShiftA | Clip | Table | GetMSB |
|  | 8x8CU | 18 | 169 | 0 | 0 | 6 | 96 | 0 | 0 |
|  | 16x16CU | 5 | 130 | 0 | 0 | 2 | 96 | 0 | 0 |
|  | 32x32CU | 1 | 113 | 0 | 0 | 0 | 96 | 0 | 0 |
|  | 64x64CU | 0 | 104 | 0 | 0 | 0 | 96 | 0 | 0 |
| **Test1.1+additional simplification** | Summary | Comparison | Add/Sub | Mul | ShiftC | ShiftA | Clip | Table | GetMSB |
|  | 8x8CU | 8 | 107 | 85 | 67 | 7 | 65 | 1 | 1 |
|  | 16x16CU | 6 | 136 | 116 | 98 | 5 | 97 | 1 | 1 |
|  | 32x32CU | 2 | 114 | 105 | 97 | 1 | 96 | 0 | 0 |
|  | 64x64CU | 0 | 105 | 100 | 96 | 0 | 96 | 0 | 0 |

If we choose the worst case (8x8 CU in HTM70 and 16x16 CU in proposal), the comparison is shown in Table 4 and Table 5.

Table 4. Number of operations in worst case (compared to HTM70)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | worst case | Comparison | Add/Sub | Mul | ShiftC | ShiftA | Clip | Table | GetMSB |
| Test 1.1 | 8x8CU | 67% | 97% | 102% | 85% | 17% | 100% | 100% | 33% |
| Test 1.3 | 8x8CU | 50% | 64% | 0% | 0% | 5% | 97% | 0% | 0% |
| Proposal  (Test1.1+additional simplification) | 16x16CU | 22% | 40% | 49% | 80% | 6% | 66% | 33% | 11% |

Table 5. Comparison of worst case

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test 1.1** | **Test 1.3** | **Proposal**  (Test 1.1 + additional simplification) |
| Comparison | to 2/3 | to 1/2 | to 1/5 |
| Add / Mul | no change | to 2/3 | to 1/2 |
| Shift | no change (overall) | removed (5% remain) | to 6/7 |
| Clip | no change | no change | to 1/3 |
| Table | no change | removed | to 1/3 |
| GetMSB | to 1/3 | removed | to 1/9 |

# Conclusion

A scheme consists of CE4 test1.1 and additional simplification is proposed. The experimental result reportedly shows 0.2 % gain in video, total video and synthesis respectively. Because this proposal simplifies the illumination process significantly and achieves coding efficiency improvement, it is recommended to adopt this technique in next 3D-HEVC WD and HTM.

# References

[1] T. Ikai, “3D-CE5.h related: Illumination compensation regression improvement and simplification” JCT3V-D0061, JCT3V 4th Meeting: Incheon, KR, 20–26 Apr. 2013

[1] T. Ikai, “CE4: Illumination compensation regression improvement and simplification” JCT3V-E0045, JCT3V 5th Meeting: Vienna, AT, 27 July – 2 Aug. 2013

# Patent rights declaration(s)

**Sharp Corporation may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**