|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  15th Meeting: Geneva, CH, 23 Oct. – 1 Nov. 2013 | Document: JCTVC-O0078 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **SCE3: performance and complexity test for cross-color inter-layer filter** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Xiang Li Jianle Chen Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121, USA  E. Alshina, A. Alshin, Y. Cho 416, Maetan 3-dong, Yeongtong-gu, Suwon, Korea, 443-742  Jie Dong, Yan Ye, Yuwen He  9710 Scranton Rd, Ste 250,  San Diego, CA 92121, USA | Tel: Email:  Tel: Email:  Tel: Email: | +1-858-651-8028  [lxiang@qti.qualcomm.com](mailto:lxiang@qti.qualcomm.com)  [cjianle@qti.qualcomm.com](mailto:cjianle@qti.qualcomm.com) [martak@qti.qualcomm.com](mailto:martak@qti.qualcomm.com)  +82 10 3026 1305 [elena\_a.alshina@samsung.com](mailto:elena_a.alshina@samsung.com)  alexander\_b.alshin@samsung.com  +1 858.210.4809 jie.dong@interdigital.com |
| *Source:* | Qualcomm Incorporated, Samsung Electronics, Ltd. & InterDigital Communications, Inc. | | |

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

# Abstract

This contribution presents performance and complexity measurement results for region based cross-color inter-layer filter. In average 0.7%(Y)/ 11.3% (U) /20.9%(V) and 0.5%(Y)/ 10.0% (U) / 18.8%(V) BD-rate gain was achieved in spatial and SNR scalability correspondently. Average memory access compare to anchor is 100% for spatial and 101% for SNR scalability. There is no increment of memory access in the worst case.

# Introduction

Inter-layer cross-color filtering was proposed in [1][2] to improve the quality of chroma inter-layer prediction. High-frequency information is extracted from corresponding up-sampled luma component and is used to enhance up-sampled chroma components. In order to resolve coding delay and allow independent processing for luma and chroma components in [3] algorithm was modified to refer directly to reconstructed base layer luma component instead of up-sampled luma signal. In [4] the number of taps for inter-layer cross-color filter was reduced from 12 to 8.

Additionally to the listed above simplifications the region based inter-layer cross-color filtering was proposed in [5]. Picture split into 2p equal-size rectangular regions is allowed for better parameters adaptation. For each region of a chroma component, one set of filter parameters is signalled. The region partitioning is adaptively determined by the encoder. Region based adaptation doubles compression gain of cross-color inter-layer filter.

The tool proposed and described in detail in [5] was tested in this contribution on top of SHM3.0.1 s/w.

# Test results

In this section, the proposed method is experimentally verified under SHVC common test conditions. Performance test results are summarized in Table 1. In average 0.7%(Y)/ 11.3% (U) /20.9%(V) and 0.5%(Y)/ 10.0% (U) / 18.8%(V) BD-rate gain was achieved in spatial and SNR scalability tests correspondently. It needs to be mentioned that tool is applied for chroma only while some luma gain is observed due to the bits saving to code chroma component.

**Table 1.** Performance test results for region based cross-color inter-layer filter.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AI HEVC 2x** | | | **AI HEVC 1.5x** | | |  |  |  |
|  | Y | U | V | Y | U | V |  |  |  |
| Class A | -2,0% | -14,9% | -18,0% |  |  |  |  |  |  |
| Class B | -1,5% | -10,2% | -21,9% | -1,1% | -10,6% | -23,2% |  |  |  |
| **Overall (Test vs Ref)** | -1,6% | -11,5% | -20,8% | -1,1% | -10,6% | -23,2% |  |  |  |
| **Overall (Test vs single layer)** | 10,9% | 1,9% | -8,0% | 9,3% | -1,5% | -14,1% |  |  |  |
| **Overall (Ref vs single layer)** | 12,8% | 14,9% | 14,6% | 10,5% | 9,8% | 9,3% |  |  |  |
| **EL only (Test vs Ref)** | -2,9% | -13,6% | -23,1% | -3,1% | -14,6% | -28,6% |  |  |  |
| Enc Time[%] | 101,6% | | | 95,9% | | |  |  |  |
| Dec Time[%] | 104,7% | | | 100,4% | | |  |  |  |
| Enc Mem[%] | 100,1% | | | 100,1% | | |  |  |  |
| BL Match | Matched | | | Matched | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  | **RA HEVC 2x** | | | **RA HEVC 1.5x** | | | **RA HEVC SNR** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A | -1,0% | -21,0% | -22,0% |  |  |  | -0,7% | -14,1% | -17,8% |
| Class B | -0,7% | -12,1% | -23,3% | -0,4% | -12,8% | -24,8% | -0,5% | -11,1% | -22,6% |
| **Overall (Test vs Ref)** | -0,8% | -14,6% | -23,0% | -0,4% | -12,8% | -24,8% | -0,6% | -12,0% | -21,3% |
| **Overall (Test vs single layer)** | 18,1% | 13,2% | 2,5% | 15,7% | 12,6% | -1,4% | 13,7% | 16,2% | 6,7% |
| **Overall (Ref vs single layer)** | 19,0% | 33,1% | 31,8% | 16,2% | 28,9% | 29,2% | 14,4% | 32,1% | 34,2% |
| **EL only (Test vs Ref)** | -1,3% | -15,8% | -24,3% | -1,0% | -14,7% | -27,4% | -1,0% | -13,3% | -22,9% |
| Enc Time[%] | 107,3% | | | 107,0% | | | 107,0% | | |
| Dec Time[%] | 117,0% | | | 115,7% | | | 116,3% | | |
| Enc Mem[%] | 100,1% | | | 100,1% | | | 107,9% | | |
| BL Match | Matched | | | Matched | | | Matched | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **LD-B HEVC 2x** | | | **LD-B HEVC 1.5x** | | | **LD-B HEVC SNR** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A | -0,5% | -18,6% | -19,0% |  |  |  | -0,5% | -13,1% | -17,1% |
| Class B | -0,3% | -7,8% | -15,9% | -0,2% | -10,0% | -21,8% | -0,3% | -7,9% | -18,9% |
| **Overall (Test vs Ref)** | -0,4% | -10,9% | -16,8% | -0,2% | -10,0% | -21,8% | -0,4% | -9,4% | -18,4% |
| **Overall (Test vs single layer)** | 27,8% | 22,9% | 16,3% | 24,5% | 19,8% | 7,0% | 23,8% | 21,8% | 13,8% |
| **Overall (Ref vs single layer)** | 28,3% | 38,9% | 39,7% | 24,8% | 33,2% | 36,0% | 24,3% | 34,7% | 39,5% |
| **EL only (Test vs Ref)** | -0,6% | -11,4% | -17,2% | -0,5% | -11,0% | -22,9% | -0,6% | -10,2% | -19,3% |
| Enc Time[%] | 102,7% | | | 103,8% | | | 102,9% | | |
| Dec Time[%] | 112,9% | | | 111,9% | | | 112,5% | | |
| Enc Mem[%] | 100,1% | | | 100,1% | | | 107,9% | | |
| BL Match | Matched | | | Matched | | | Matched | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| **Optional Tests** | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  | **LD-P HEVC 2x** | | | **LD-P HEVC 1.5x** | | | **LD-P HEVC SNR** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A | -0,5% | -17,6% | -18,0% |  |  |  | -0,4% | -11,9% | -15,6% |
| Class B | -0,3% | -7,3% | -15,0% | -0,3% | -9,4% | -20,8% | -0,3% | -7,5% | -17,4% |
| **Overall (Test vs Ref)** | -0,4% | -10,2% | -15,9% | -0,3% | -9,4% | -20,8% | -0,4% | -8,8% | -16,9% |
| **Overall (Test vs single layer)** | 26,0% | 23,1% | 16,8% | 22,5% | 20,3% | 7,9% | 23,0% | 22,5% | 15,5% |
| **Overall (Ref vs single layer)** | 26,5% | 37,9% | 38,9% | 22,8% | 32,8% | 35,6% | 23,4% | 34,6% | 39,3% |
| **EL only (Test vs Ref)** | -0,6% | -10,8% | -16,3% | -0,5% | -10,4% | -22,1% | -0,6% | -9,5% | -17,9% |
| Enc Time[%] | 100,5% | | | 103,0% | | | 98,5% | | |
| Dec Time[%] | 111,8% | | | 112,7% | | | 109,4% | | |
| Enc Mem[%] | 100,1% | | | 100,1% | | | 107,9% | | |
| BL Match | Matched | | | Matched | | | Matched | | |

Complexity assessment was performed using AhG17 materials. Memory access in the worst case for tested variant of re-sampling filter is identical to anchor. Average memory access compare to anchor is 100% for spatial and 101% for SNR scalability. Number of multiplication s 2-5% higher for motion compensation test scenario and 17-18% higher in all-intra test. Results of complexity assessment are presented in Table 2.

**Table 2.** Average memory access compare to anchor

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **AI HEVC 2x** | | |  | **AI HEVC 1,5x** | | |  |  |  | |  |
|  | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults |  |  | |  |
| Class A | 101% | 99% | 99% | 117% |  |  |  |  |  |  | |  |
| Class B | 102% | 100% | 100% | 119% | 103% | 100% | 101% | 117% |  |  |  |  |
| Overall | **101%** | **100%** | **100%** | **118%** | **103%** | **100%** | **101%** | **117%** |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **RA HEVC 2x** | | |  | **RA HEVC 1.5x** | | |  | **RA HEVC SNR** | | |  |
|  | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults |
| Class A | 100% | 100% | 100% | 103% |  |  |  |  | 100% | 101% | 101% | 103% |
| Class B | 100% | 100% | 100% | 102% | 101% | 100% | 100% | 103% | 100% | 101% | 101% | 102% |
| Overall | **100%** | **100%** | **100%** | **103%** | **101%** | **100%** | **100%** | **103%** | **100%** | **101%** | **101%** | **102%** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **LD-P HEVC 2x** | | |  | **LD-P HEVC 1.5x** | | |  | **LD-P HEVC SNR** | | |  |
|  | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults |
| Class A | 100% | 100% | 100% | 105% |  |  |  |  | 101% | 101% | 101% | 104% |
| Class B | 100% | 100% | 100% | 103% | 101% | 100% | 100% | 105% | 100% | 101% | 101% | 103% |
| Overall | **100%** | **100%** | **100%** | **104%** | **101%** | **100%** | **100%** | **105%** | **101%** | **101%** | **101%** | **104%** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **LD\_B HEVC 2x** | | |  | **LD-B HEVC 1.5x** | | |  | **LD-B HEVC SNR** | | |  |
|  | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults | Pure | DDR2 | DDR3 | Mults |
| Class A | 100% | 100% | 100% | 104% |  |  |  |  | 101% | 101% | 101% | 103% |
| Class B | 100% | 100% | 100% | 103% | 100% | 100% | 100% | 104% | 100% | 101% | 101% | 102% |
| Overall | **100%** | **100%** | **100%** | **104%** | **100%** | **100%** | **100%** | **104%** | **100%** | **101%** | **101%** | **103%** |

The estimation of memory access and multiplications number in the worst case was also the part on SCE3. Results of the worst case complexity analysis for proposed algorithm compare to anchor are shown in Table 3.

Even if proposed algorithm is included into the inter-layer processing both memory access and multiplications number of inter-layer texture prediction do not exceed complexity of motion compensation bi-prediction. So the worst case SHVC complexity would be the same even if this tool is added.

Proposed algorithm requires additional step for Chroma processing compare to SHM3.0 re-sampling. But Luma and Chroma are re-sampled independently. Amount of Chroma samples to process is 4 time smaller than Luma and re-sampling filter for Chroma is twice shorter compare to Luma. In terms of number of multiplication operations Chroma re-sampling is ~1/4 of Luma re-sampling (per 1 sample inter-layer texture prediction 7 mults are needed for Chroma re-sampling vs 30 – for Luma). With proposed algorithm amount of operations for Luma is un-changed. The number of mults in Chroma inter-layer processing goes from 7 to 11 and still ~1/3 compare to operations in Luma re-sampling part. Based on these data we can conclude that with proper implementation adding proposed algorithm to SHVC design will not introduce additional latency in inter-layer processing part.

**Table 3.** Worst case complexity analysis (spatial scalability ×1,5).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Memory access per sample prediction | | | | Multiplications per sample prediction | | | |
|  | MC bi-pred | | Inter-layer process | | MC bi-pred | | Inter-layer process | |
|  | Luma | Chroma | Luma | Chroma | Luma | Chroma | Luma | Chroma |
| SHM3.0 | **12** | **6** | 10,5 | 4 | **46** | **11** | 30 | 7 |
| Proposed | **12** | **6** | 12 | 4 | **46** | **11** | 30 | 11 |

Authors would like to thank Huawei for the cross-check of presented results.

# Conclusion

Based on test results shown:

* 0.7%(Y)/ 11.3% (U) /20.9%(V) and 0.5%(Y)/ 10.0% (U) / 18.8%(V) BD-rate gain in spatial and SNR scalability tests correspondently;
* average memory access compare to anchor is 100% for spatial and 101% for SNR scalability.

Qualcomm, Samsung and InterDigital propose inclusion of region based cross-color inter-layer filter into the next release of SHVC draft text and reference software.

# References

1. J. Dong, Y. He, Y. Ye, “Chroma enhancement for ILR picture”, JCTVC-L0059, Geneva, Switzerland, Jan. 2013.
2. J. Dong, Y. Ye, Y. He, “Chroma enhancement for ILR picture,” JCTVC-M0183, Incheon, Korean, 18–26 Apr. 2013
3. E. Alshina, A.Alshina, Y.Cho, “Non SCE4: simplified design of cross-color inter-layer (test 4.2.4),” JCTVC-M0089, Incheon, Korean, 18–26 Apr. 2013.
4. X. Li, J. Chen, W. Pu, M. Karczewicz, “Non-SCE4: Simplification of chroma enhancement for inter layer reference picture generation”, JCTVC-M0253, Incheon, Korean, 18–26 Apr. 2013
5. Xiang Li, Wei Pu, Jianle Chen, Marta Karczewicz, E. Alshina, A.Alshina, Y.Cho “Non-SCE3: Region based Inter-layer Cross-Color Filtering”, JCTVC-N0229, Vienna, Austria, July 2013

# Patent rights declarations

**Qualcomm Incorporated 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).**

**Samsung Electronics, Ltd. 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).**

**InterDigital Communications, Inc. 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).**

# Specification draft text

## Sequence parameter set extension syntax and semantics

|  |  |
| --- | --- |
| sps\_extension( ) { | **Descriptor** |
| … |  |
| **sps\_inter\_layer\_cross\_color\_enable\_flag** | u(1) |
| if(sps\_inter\_layer\_cross\_color\_enable\_flag)**{** |  |
| **sps\_max\_inter\_layer\_cross\_color\_part\_log4** | ue(v) |
| } |  |
| } |  |

**sps\_inter\_layer\_cross\_color\_enable\_flag**equal to1 specifies that inter-layer cross-color filteringis enabled. sps\_inter\_layer\_cross\_color\_enable\_flagequal to0 specifies that inter-layer cross-color filtering is disabled. When not present, sps\_inter\_layer\_cross\_color\_enable\_flagis inferred to be equal to 0.

**sps\_max\_inter\_layer\_cross\_color\_part\_log4**specifies the maximal number of paritions used in inter-layer cross-color filtering. When not present, sps\_max\_inter\_layer\_cross\_color\_part\_log4 is inferred to be equal to 0.

## APS

|  |  |
| --- | --- |
| aps\_rbsp( ) { | Descriptor |
| **aps\_id** | ue(v) |
| for(uv=0;uv<2;uv++){ |  |
| **pic\_inter\_layer\_cross\_color\_part\_log4**[uv] | ue(v) |
| for(i=0;i<PicInterLayerCrossColorPart[uv];i++) |  |
| inter\_layer\_cross\_color\_filter\_param(uv,i) |  |
| } |  |
| } |  |

**aps\_id** identifies the adaptation parameter set that is referred to in the slice segment header. The value of aps\_id shall be in the range of 0 to TBD, inclusive.

**pic\_inter\_layer\_cross\_color\_part\_log4**[uv] specifies variable PicInterLayerCrossColorPart[uv] which is the number of partitions used in inter-layer corss-color filtering for the chroma component in the current picture. pic\_inter\_layer\_cross\_color\_part\_log4[uv] shall be between 0 and sps\_max\_inter\_layer\_cross\_color\_part\_log4, inclusively. When not present, pic\_inter\_layer\_cross\_color\_part\_log4[uv] is inferred to be equal to 0.

The NAL unit type of proposed APS NAL unit type is 41 (currently reserved as RSV\_NVCL41).

## General slice segment header syntax and semantics



|  |  |
| --- | --- |
| slice\_segment\_header( ) { | Descriptor |
| **first\_slice\_segment\_in\_pic\_flag** | u(1) |
| …… |  |
| if(sps\_inter\_layer\_cross\_color\_enable\_flag) |  |
| **slice\_aps\_id** | ue(v) |
| …… |  |
| } |  |

**slice\_aps\_id** identifies the adaptation parameter set that contains the chroma filter coefficients for the current slice. The value of slice\_aps\_id shall be in the range of 0 to TBD, inclusive.

## inter\_layer\_cross\_color\_filter\_param

|  |  |
| --- | --- |
| inter\_layer\_cross\_color\_filter\_param(uv,i){ |  |
| **inter\_layer\_cross\_color\_flag**[uv][i] | u(1) |
| if (inter\_layer\_cross\_color\_flag[uv][i]){ |  |
| for(j=0; j<7; j++) |  |
| **inter\_layer\_cross\_color\_filter\_coeff\_ plus8**[uv][i][j] | u(4) |
| **inter\_layer\_cross\_color\_filter\_scaling\_factor\_abs\_minus1**[uv][i] | u(10) |
| **inter\_layer\_cross\_color\_filter\_scaling\_factor\_sign**[uv][i] | u(1) |
| } |  |
| } |  |

**inter\_layer\_cross\_color\_flag**[uv][i] equal to 1 specifies that inter-layer cross-color filtering will be applied to the ith region of the chroma component uv. inter\_layer\_cross\_color\_flag[uv][i] equal to 0 specifies that inter-layer cross-color filtering will not be applied to the ith region of the chroma component uv. When not present, inter\_layer\_cross\_color\_flag is inferred to be equal to 0.

**inter\_layer\_cross\_color\_filter\_coeff\_plus8**[uv][i][j] minus 8 specifies the jth filter coefficient of the inter-layer cross-color filter for the ith region of the chroma component uv. The value of inter\_layer\_cross\_color\_filter\_coeff\_ plus8[uv][i][j] shall be in the range of 0 to 15, inclusivly.

**inter\_layer\_cross\_color\_filter\_scaling\_factor\_abs\_minus1**[uv][i]and **inter\_layer\_cross\_color\_filter\_scaling\_factor\_sign**[uv][i]together specify the value of variable InterLayerCrossColorFilterScalingFactor[uv][i]for the ith region of the chroma component uv as follows:

InterLayerCrossColorFilterScalingFactor[uv][i]= (1 – 2 \*inter\_layer\_cross\_color\_filter\_scaling\_factor\_sign[uv][i]) \* (inter\_layer\_cross\_color\_filter\_scaling\_factor\_abs\_minus1[uv][i]+1)

The value of inter\_layer\_cross\_color\_filter\_scaling\_factor\_abs\_minus1 shall be in the range of 1 to 1023, inclusively.

## Decoding Process

**G.8.1.4 Derivation process for inter layer reference pictures**

**……**

The following steps are applied to derive the resampled inter layer reference picture rsPic.

– if PicWidthInSamplesL is equal to RefLayerPicWidthInSamplesL and PicHeightInSamplesL is equal to RefLayerPicHeightInSamplesL and the values of ScaledRefLayerLeftOffset, ScaledRefLayerTopOffset, ScaledRefLayerRightOffset and ScaledRefLayerBottomOffset are all equal to 0

* + rsPic is set equal to rlPic.
  + If sps\_inter\_layer\_cross\_color\_enable\_flag is equal to 1, the chroma enhancement filtering process as specified in subclause H.8.1.4.3 is invoked with the sample values of rsPicSample as input, and with the modifed chroma sample values of rsPicSample as output.

– otherwise, rsPic is derived as follows:

* + The PicOrderCntVal value of rsPic is set equal to the PicOrderCntVal value of rlPic.
  + When SamplePredEnabledFlag[ currLayerId ][ rLId ] is equal to 1, the follows apply:
    - The picture sample resampling process as specified in subclause H.8.1.4.1 is invoked with the sample arrays of reference layer picture rlPic as input, and with the sample arrays of resampled picture rsPic as output.
    - When sps\_inter\_layer\_cross\_color\_enable\_flag is equal to 1, the chroma enhancement filtering process as specified in subclause H.8.1.4.3 is invoked with the luma sample values of rlPicSample and chroma sample values of rsPicSample as input, and with the modifed chroma sample values of rsPicSample as output.
  + When MotionPredEnabledFlag[ currLayerId ][ rLId ] is equal to 1, the picture motion field resampling process as specified in subclause H.8.1.4.2 is invoked with reference layer picture rlPic and its motion field as inputs, and with the motion field of resampled picture rsPic as output.

**……**

**G.8.1.4.3 Chroma enhancement filtering process for inter layer reference pictures**

Input to this process is:

– a ( RefLayerPicWidthInSamplesL ) x ( RefLayerPicHeightInSamplesL ) array rlPicSampleL of luma samples.

– a ( PicWidthInSamplesC ) x ( PicHeightInSamplesC ) array rsPicSampleCb of chroma samples of the component Cb

– a ( PicWidthInSamplesC ) x ( PicHeightInSamplesC ) array rsPicSampleCr of chroma samples of the component Cr

Outputs of this process are:

– a modified ( PicWidthInSamplesC ) x ( PicHeightInSamplesC )array rsPicSampleCb of chroma samples of the component Cb,

– a modified ( PicWidthInSamplesC ) x ( PicHeightInSamplesC )array rsPicSampleCr of chroma samples of the component Cr.

For each chroma component, the following process applies

Variable PicInterLayerCrossColorParts[uv] with uv=0 indicating Cb and uv=1 indicating Cr, are derived as PicInterLayerCrossColorParts[uv] = 4 ^ pic\_inter\_layer\_cross\_color\_part\_log4[uv]

The color plane is equally splitted into PicInterLayerCrossColorParts[uv] rectangular regions. The regions are indexed in an order of quadtree style. Given a region index Idx, the top-left (X0, Y0) and bottom-right (X1, Y1) of the region is derived as follows.

StepX = ScaledRefLayerPicWidthInSamplesC >> pic\_inter\_layer\_cross\_color\_part\_log4[uv]

StepY = ScaledRefLayerPicHeightInSamplesC >> pic\_inter\_layer\_cross\_color\_part\_log4[uv]

X0 = 0

Y0 = 0

for(Depth = 1 ; Depth <= pic\_inter\_layer\_cross\_color\_part\_log4[uv] ; Depth++ )

{

DepthIdx = Idx >> ( ( MaxDepth - Depth ) << 1 )

DepthIdx &= 0x03

X0 += ( DepthIdx & 0x01 ) \* (ScaledRefLayerPicWidthInSamplesC >> Depth )

Y0 += ( DepthIdx >> 1 ) \* (ScaledRefLayerPicHeightInSamplesC >> Depth )

}

X1 = X0 + StepX

Y1 = Y0 + StepY

Where MaxDepth is the maximum supported depth. It may be fixed to 3.

If inter\_layer\_cross\_color\_flag[uv][i] is equal to 1, the modified array rsPicSampleCb or rsPicSampleCr of the samples of the chroma component is derived by invoking the enhancing process of chroma sample values specified in subclause G.8.1.4.3.1 with the (2\*X1-2\*X0)x(2\*Y1-2\*Y0) array rlPicSampleL, (X1-X0)x(Y1-Y0) array rsPicSampleCb or rsPicSampleCr, inter\_layer\_cross\_color\_filter\_coeff\_plus8[uv][Idx][j] (j=0…6), inter\_layer\_cross\_color\_filter\_scaling\_factor\_abs\_minus1[uv][Idx], and inter\_layer\_cross\_color\_filter\_scaling\_factor\_sign[uv][Idx] given as inputs, and with the modified ((X1-X0))x((Y1-Y0)) array rsPicSampleCb or rsPicSampleCr as output. If inter\_layer\_cross\_color\_flag[uv][i] is equal to 0, the array rsPicSampleCb or rsPicSampleCr of the samples of the chroma component is not modified.

**G.8.1.4.3.1 Enhancing process of chroma sample values**

Inputs to this process are:

– a luma reference sample array rlPicSampleL.,

– a (X0,Y0) ~ (X1,Y1) array rsPicSampleC,

– a variable array filter\_coeff\_plus8 [i] ( i = 0…6 ),

– a variable ScalingFactor and the sign of ScalingFactor,

Outputs of this process is:

– a modified (X0,Y0) ~ (X1,Y1) array rsPicSampleC.

The modified chroma samples rsPicSampleC[ xPC ][ yPC ] with ( xPC = X0…X1 – 1, yPC = Y0…Y1 – 1) are derived by applying the following ordered steps:

1. Temporal array filter\_coeff [i] ( i = 0…7 ) values are specified as follows:

filter\_coeff [i] = filter\_coeff\_plus8 [i] +8( i = 0…6 );

filter\_coeff [7] =- filter\_coeff [6]- filter\_coeff [5]- filter\_coeff [4]- filter\_coeff [3]- filter\_coeff [2]- filter\_coeff [1]- filter\_coeff [0];

1. The variable Temp is derived as follows:

refW      = RefLayerPicWidthInSamplesL  
refH      = RefLayerPicHeightInSamplesL

Temp=

( filter\_coeff [0] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL ), Clip3(0, refH -1, yPL -1)] )  
+ ( filter\_coeff [1] ) \*( rlPicSampleL [ Clip3(0, refW -1, xPL -1), Clip3(0, refH -1, yPL )] )  
+ ( filter\_coeff [2] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL ), Clip3(0, refH -1, yPL )] )  
+ ( filter\_coeff [3] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL +1), Clip3(0, refH -1, yPL )] )  
+ ( filter\_coeff [4] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL -1), Clip3(0, refH -1, yPL +1)] )  
+ ( filter\_coeff [5] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL), Clip3(0, refH -1, yPL +1)] )  
+ ( filter\_coeff [6] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL +1), Clip3(0, refH -1, yPL +1)] )  
+ ( filter\_coeff [7] ) \* (rlPicSampleL [ Clip3(0, refW -1, xPL ), Clip3(0, refH -1, yPL +2)] )

here

xPL = (((2\*xPC ‑ offsetX )\* scaleFactorX  + ( 1 << ( shiftX − 1 ) ) ) >> shiftX+8)>>4

yPL = (((2\* yPC ‑ offsetY)\* scaleFactorY  + ( 1 << ( shiftY − 1 ) ) ) >> shiftY+8)>>4

1. The variable ScaledTemp is derived as follows:

ScaledTemp=Sign(temp \* ScalingFactor) \* ((Abs(temp \* ScalingFactor) +8192) >> 14)

1. The modified rsPicSampleC[ xPC ][ yPC ] is derived as follows:

rsPicSampleC[ xPC ][ yPC ] = Clip3( 0, ( 1 << BitDepthC) – 1, rsPicSampleC[ xPC ][ yPC ]+ScaledTemp).