|  |  |
| --- | --- |
| **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-I0088 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Simplification of filtering in DBBP and its extension to PU boundary** | | |
| *Status:* | Input document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jin Young Lee, Mikhail Mishurovskiy, Min Woo Park, Yongjin Cho, and Chanyul Kim | Email: | [jinyoung79.lee@samsung.com](mailto:jinyoung79.lee@samsung.com) |
| *Source:* | Samsung Electronics Co. Ltd. | | |

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

# Proposed Text

Method 1 – Simplification of filtering in DBBP

I. 8.5.3.3.9 Decoding process for depth based block partition wise inter sample prediction

...

When partIdx is equal to 1, the array~~s~~ PredSamplesDbbpL~~,~~ ~~PredSamplesDbbp~~~~Cb~~ ~~and PredSamplesDbbp~~~~Cr~~ ~~are~~ is modified as follows:

* The derivation process for contour boundary filtered samples as specified in subclause I.8.5.3.3.9.1 is invoked with, the luma coding block size block nCbSL, ~~the current coding block size nCbS~~~~X~~ ~~set equal to nCbS~~~~L~~~~,~~ the array segMask, the array predSamples of prediction samples equal to PredSamplesDbbpL as inputs and the output is assigned to the array PredSamplesDbbpL of luma prediction samples.
* ~~The derivation process for contour boundary filtered samples as specified in subclause  is invoked with, the luma coding block size block nCbS~~~~L~~~~, the current coding block size nCbS~~~~X~~ ~~set equal to nCbS~~~~C~~~~, the array segMask, the array predSamples of prediction samples equal to PredSamplesDbbp~~~~Cb~~ ~~as inputs and the output is assigned to the array PredSamplesDbbp~~~~Cb~~ ~~of luma prediction samples.~~
* ~~The derivation process for contour boundary filtered samples as specified in subclause  is invoked with, the luma coding block size block nCbS~~~~L~~~~, the current coding block size nCbS~~~~X~~ ~~set equal to nCbS~~~~C~~~~, the array segMask, the array predSamples of prediction samples equal to PredSamplesDbbp~~~~Cr~~ ~~as inputs and the output is assigned to the array PredSamplesDbbp~~~~Cr~~ ~~of luma prediction samples.~~

I. 8.5.3.3.9.1 Derivation process for contour boundary filtered samples

Inputs to this process are:

* a variable nCbSL specifying the size of the current luma coding block,
* ~~a variable nCbS~~~~X~~ ~~specifying the size of the current coding block,~~
* an (nCbSL)x(nCbSL) array segMask
* an ~~(nCbS~~~~X~~~~)x(nCbS~~~~X~~~~)~~ (nCbSL)x(nCbSL) array predSamples prediction samples

Outputs to this process are:

* an modified ~~(nCbS)x(nCbS)~~ (nCbSL)x(nCbSL) array predSamples of luma prediction samples

~~The (nCbS~~~~X~~~~)x(nCbS~~~~X~~~~) array p is set equal to predSamples and the variable n is set equal to ( nCbS~~~~L~~~~/ nCbS~~~~X~~~~).~~

The values of predSamples are derived as specified in the following:

for ( y = 0; y < ~~nCbS~~~~X~~ nCbSL; y++ )  
 for( x = 0; x < ~~nCbS~~~~X~~ nCbSL; x++ ) {  
 tFlag = segMask[ n \* x ][ Max( 0, n \* ( y − 1 ) ) ]  
 lFlag = segMask[ Max( 0, (n \* ( x − 1 ) ) ][ n \* y ]  
 bFlag = segMask[ n \* x ][ Min( n \* ( y + 1 ), nCbSL − 1 ) ]  
 rFlag = segMask[ Min( n \* ( x + 1 ), nCbSL − 1 ) ][ n\*y ]  
 ~~cFlag = segMask[ n \* x ][ n \* y ]~~  
 filt = p[ x ][ y ]  
 if( lFlag != rFlag ~~( lFlag | | cFlag | | rFlag ) && ( !lFlag | | !cFlag | | !rFlag )~~ )  
 filt = ( p[ Max( 0, x − 1 ) ][ y ] + ( filt << 1 ) + p[ Min( x + 1, nCbSX − 1 ) ][ y ] ) >> 2  
 if(tFlag != bFlag ~~( tFlag | | cFlag | | bFlag ) && ( !tFlag | | !cFlag | | !bFlag )~~ )  
 filt = ( p[ x ][ Max( 0, y − 1 ) ] + ( filt << 1 ) + p[ x ][ Min( y + 1, nCbSX − 1 ) ] ) >> 2  
 predSamples[ x ][ y ] = filt  
 }

Method 2 – Extension of the filtering to PU boundary

I.8.5.3.1 General

...

3. When partIdx is not equal to 0, dbbp\_flag[xCb][yCb] is equal to 0, DepthFlag is equal to 0, and nuh\_layer\_id is not equal to 0, the derivation process for pu boundary filtered samples as specified in subclause  is invoked with, the luma coding block size block nCbSL, the array predSamples of prediction samples equal to PredSamplesL, the luma location ( xB1, yB1 ), and the variables nPbW and nPbH as inputs and the output is assigned to the modified array PredSamplesL of the luma prediction samples.

…

I. 8.5.3.3.9.2 Derivation process for pu boundary filtered samples

Inputs to this process are:

* a variable nCbSL specifying the size of the current luma coding block,
* an (nCbSL)x(nCbSL) array predSamples prediction samples,
* a luma location ( xBl, yBl ) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block,
* a variable nPbW specifying the width of the current luma prediction block,
* a variable nPbH specifying the width of the current luma prediction block

Outputs to this process are:

* an modified (nCbSL)x(nCbSL) array predSamples of luma prediction samples

The (nCbSL)x(nCbSL) arrays p, q, and r are set equal to predSamples.

The values of predSamples are modified as the following ordered steps:

1. When yBl is not equal to 0, the values of r is modified as specified in the following:

for ( y = yB1-1; y < = yB1; y++ )  
 for( x = xB1; x < xB1+ nPbW; x++ ) {  
 filt = p[ x ][ y ]  
 filt = ( p[ x ][ Max( 0, y − 1 ) ] + ( filt << 1 ) + p[ x ][ Min( y + 1, nCbSX − 1 ) ] ) >> 2  
 q[ x ][ y ] = filt  
 }

2. When xBl is not equal to 0, the values of q are modified as specified in the following:

for ( y = yB1; y < yB1+nPbH; y++ )  
 for( x = xB1-1; x <= xB1; x++ ) {  
 filt = q[ x ][ y ]  
 filt = ( q[ Max( 0, x − 1 ) ][ y ] + ( filt << 1 ) + q[ Min( x + 1, nCbSX − 1 ) ][ y ] ) >> 2  
 r[ x ][ y ] = filt  
 }

3. the values of predSamples are modified as specified in the following:

for ( y = 0; y < nCbSL; y++ )  
 for( x = 0; x < nCbSL; x++ )   
 predSamples[ x ][ y ] = r[x][y]

# Patent Rights Declaration(s)

**Samsung Electronics Co. Ltd. may have IPR 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).**