|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extension Development**  **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- E0157 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE6 related: Applying Depth Look-up Table to Intra Modes of Depth Map** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Hongbin Liu ([hongbin.liu@lge.com](mailto:hongbin.liu@lge.com))  Jie Jia ([jie.jia@lge.com](mailto:jie.jia@lge.com)) |  |  |
| *Source:* | LG Electronics | | |

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

Spec Change:

**H.7.3.9.1 General Coding unit syntax**

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize , ctDepth) { | **Descriptor** |
| if( transquant\_bypass\_enable\_flag ) { |  |
| **cu\_transquant\_bypass\_flag** | ae(v) |
| } |  |
| if( slice\_type != I ) |  |
| **skip\_flag**[ x0 ][ y0 ] | ae(v) |
| if( skip\_flag[ x0 ][ y0 ] ) { |  |
| prediction\_unit( x0, y0, log2CbSize ) |  |
| if ( iv\_res\_pred\_flag[ nuh\_layer\_id ] && TempRefPicInListsFlag ) |  |
| iv\_res\_pred\_weight\_idx | ae(v) |
| if ( icEnableFlag ) |  |
| **ic\_flag** | ae(v) |
| } |  |
| else { |  |
| nCbS = ( 1 << log2CbSize ) |  |
| if( slice\_type != I ) |  |
| **pred\_mode\_flag** | ae(v) |
| if( ( PredMode[ x0 ][ y0 ] ! = MODE\_INTRA | | log2CbSize = = Log2MinCbSize ) &&  !predPartModeFlag) |  |
| **part\_mode** | ae(v) |
| if( PredMode[ x0 ][ y0 ] = = MODE\_INTRA ) { |  |
| if( PartMode = = PART\_2Nx2N && pcm\_enabled\_flag &&  log2CbSize >= Log2MinIPCMCUSize &&  log2CbSize <= Log2MaxIPCMCUSize ) |  |
| **pcm\_flag** | ae(v) |
| if( pcm\_flag ) { |  |
| **num\_subsequent\_pcm** | tu(3) |
| NumPCMBlock = num\_subsequent\_pcm + 1 |  |
| while( !byte\_aligned( ) ) |  |
| **pcm\_alignment\_zero\_bit** | f(1) |
| pcm\_sample( x0, y0, log2CbSize ) |  |
| } else { |  |
| pbOffset = ( PartMode = = PART\_NxN ) ? ( nCbS / 2 ) : 0 |  |
| log2PbSize = log2CbSize − ( PartMode = = PART\_NxN ? 1 : 0 ) |  |
| for( j = 0; j <= pbOffset; j = j + pbOffset ) |  |
| for( i = 0; i <= pbOffset; i = i + pbOffset ) { |  |
| if( vps\_depth\_modes\_flag[ nuh\_layer\_id ] ) |  |
| depth\_mode\_parameters( x0 + i ,  y0+ j ) |  |
| if( depth\_intra\_mode[ x0 + i ][ y0 + j ] = = INTRA\_DEP\_NONE ) |  |
| **prev\_intra\_luma\_pred\_flag**[ x0 + i ][ y0+ j ] | ae(v) |
| } |  |
| for( j = 0; j <= pbOffset; j = j + pbOffset ) |  |
| for( i = 0; i <= pbOffset; i = i + pbOffset ) { |  |
| if( depth\_intra\_mode[ x0 + i ][ y0 + j ] = = INTRA\_DEP\_NONE) { |  |
| if( prev\_intra\_luma\_pred\_flag[ x0 + i ][ y0+ j ] ) |  |
| **mpm\_idx**[ x0 + i ][ y0+ j ] | ae(v) |
| else |  |
| **rem\_intra\_luma\_pred\_mode**[ x0 + i ][ y0+ j ] | ae(v) |
| } |  |
| } |  |
| if ( !SdcFlag[ x0 ][ y0 ] ) |  |
| **intra\_chroma\_pred\_mode**[ x0 ][ y0 ] | ae(v) |
| if ( DepthFlag && dlt\_flag[ nuh\_layer\_id ] && !SdcFlag[ x0 ][ y0 ] ) |  |
| **residual\_dlt\_index\_flag** | ae(v) |
| } |  |
| } else { |  |
| if( PartMode = = PART\_2Nx2N ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS ) |  |
| if ( iv\_res\_pred\_flag[ nuh\_layer\_id ] && TempRefPicInListsFlag ) |  |
| **iv\_res\_pred\_weight\_idx** | ae(v) |
| } else if( PartMode = = PART\_2NxN ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS / 2 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 2 ), nCbS, nCbS / 2 ) |  |
| } else if( PartMode = = PART\_Nx2N ) { |  |
| prediction\_unit( x0, y0, nCbS / 2, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0, nCbS / 2, nCbS ) |  |
| } else if( PartMode = = PART\_2NxnU ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS / 4 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 4 ), nCbS, nCbS \*3 / 4 ) |  |
| } else if( PartMode = = PART\_2NxnD ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS \*3 / 4 ) |  |
| prediction\_unit( x0, y0 + ( nCbS \* 3 / 4 ), nCbS, nCbS / 4 ) |  |
| } else if( PartMode = = PART\_nLx2N ) { |  |
| prediction\_unit( x0, y0, nCbS /4, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS / 4 ), y0, nCbS \*3 / 4, nCbS) |  |
| } else if( PartMode = = PART\_nRx2N ) { |  |
| prediction\_unit( x0, y0, nCbS \*3 / 4, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS \* 3 / 4 ), y0, nCbS / 4, nCbS ) |  |
| } else { /\* PART\_NxN \*/ |  |
| prediction\_unit( x0, y0, nCbS / 2, nCbS / 2) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0, nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 2 ), nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), nCbS / 2, nCbS / 2 ) |  |
| } |  |
| } |  |
| if ( icEnableFlag ) |  |
| **ic\_flag** | ae(v) |
| if( !pcm\_flag ) { |  |
| if( PredMode[ x0 ][ y0 ] != MODE\_INTRA &&   !(PartMode = = PART\_2Nx2N && merge\_flag[x0][y0]) ) |  |
| **no\_residual\_syntax\_flag** | ae(v) |
| if( !no\_residual\_syntax\_flag ) { |  |
| MaxTrafoDepth = ( PredMode[ x0 ][ y0 ] = = MODE\_INTRA ?   max\_transform\_hierarchy\_depth\_intra + IntraSplitFlag :   max\_transform\_hierarchy\_depth\_inter ) |  |
| transform\_tree( x0, y0 x0, y0, log2CbSize, 0, 0 ) |  |
| } |  |
| } |  |
| } |  |
| } |  |

**H.7.4.9.1 General coding unit semantics**

**residual\_dlt\_index\_flag** equal to 1 specifies that residual of the current coding unit is the difference of depth look-up table index between original depth value and predicted depth value. residual\_dlt\_index\_flagequal to 0 specifies residual of the current coding unit is the difference between original depth value and predicted depth value. When not present, residual\_dlt\_index\_flag is inferred to be equal to 0.

**8.6.5 Picture construction process prior to in-loop filter process**

Inputs of this process are:

– a location ( xB, yB ) specifying the top-left luma sample of the current block relative to the top‑left sample of the current picture component,

– a variable nS specifying the size of the current block,

– a variable cIdx specifying the colour component of the current block,

– a (nS)x(nS) array predSamples specifying the predicted samples of the current block,

– a (nS)x(nS) array resSamples specifying the residual samples of the current block.

Depending on the colour component cIdx, the following assignments are made.

– If cIdx is equal to 0, recSamples corresponds to the reconstructed picture sample array SL and the function clipCidx1 corresponds to Clip1Y.

– Otherwise, if cIdx is equal to 1, recSamples corresponds to the reconstructed chroma sample array SCb and the function clipCidx1 corresponds to Clip1C.

– Otherwise (cIdx is equal to 2), recSamples corresponds to the reconstructed chroma sample array SCr and the function clipCidx1 corresponds to Clip1C.

– if residual\_dlt\_index\_flag is equal to 0,

The (nS)x(nS) block of the reconstructed sample array recSamples at location ( xB, yB ) is derived as follows.

recSamples[ xB+i ][ yB+j ] = clipCidx1( predSamples[ i ][ j ] + resSamples[ i ][ j ] ) (8‑272)  
 with i = 0..nS − 1, j = 0..nS − 1

– otherwise (residual\_dlt\_index\_flag is equal to 1),

The (nS)x(nS) block of the reconstructed sample array recSamples at location ( xB, yB ) is derived as follows.

recSamples[ xB+i ][ yB+j ] = Idx2DepthValue[ clip3( 0, num\_depth\_values\_in\_dlt[ nuh\_layer\_id ] – 1,

DepthValue2Idx[ predSamples[ i ][ j ] ] + resSamples[ i ][ j ] ) ] with i = 0..nS − 1, j = 0..nS – 1 (8-xxx)

**H.9.3.1.1 Initialization process for context variables**

Table H‑12 – Association of ctxIdx and syntax elements for each initializationType in the initialization process

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Syntax element** | **ctxIdxTable** | **initType** | | |
| **0** | **1** | **2** |
| coding\_unit() | dmm\_flag |  | 0 | 1 | 2 |
| dmm\_mode |  | 0 | 1 | 2 |
| wedge\_full\_tab\_idx |  | 0 | 1 | 2 |
| wedge\_predtex\_tab\_idx | Table H‑17 | 0 | 1 | 2 |
| dmm\_delta\_end\_flag dmm\_delta\_end\_abs\_minus1 |  | 0 | 1 | 2 |
| dmm\_dc\_1\_abs dmm\_dc\_2\_abs |  | 0 | 1 | 2 |
| ic\_flag |  |  | 0 | 1 |
| mvp\_l0\_idx mpv\_l1\_idx |  |  | 0..1 | 2..3 |
| sdc\_flag |  | 0..2 | 3..5 | 6..8 |
| sdc\_residual\_flag |  | 0..1 | 2..3 | 4..5 |
| sdc\_residual\_sign\_flag |  | 0 | 1 | 2 |
| sdc\_residual\_abs\_minus1 |  | 0..19 | 20..39 | 40..59 |
| sdc\_pred\_mode |  | 0..2 | 3..5 | 6..8 |
| residual\_dlt\_index\_flag | xx | 0 | 1 | 2 |

Table H‑xx – Values of variable initValue for residual\_dlt\_index\_flag ctxIdx

|  |  |  |  |
| --- | --- | --- | --- |
| **Initialization variable** | **residual\_dlt\_index\_flag** | | |
| **0** | **1** | **2** |
| **initValue** | 154 | 154 | 154 |

H.9.3.2 Binarization process

Table H‑26 – Syntax elements and associated types of binarization, maxBinIdxCtx, ctxIdxTable, and ctxIdxOffset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| sdc\_flag | 0 | FL, cMax = 1 | 0 |  | 0 |
| 1 | 0 |  | 3 |
| 2 | 0 |  | 6 |
| residual\_dlt\_index\_flag | 0 | FL, cMax = 1 | 0 | xx | 0 |
| 1 | 0 | xx | 1 |
| 2 | 0 | xx | 2 |

H.9.3.3.1 Derivation process for ctxIdx

Table H‑30 – Assignment of ctxIdxInc to syntax elements with context coded bins

| **Syntax element** | **binIdx** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | **1** | **2** | **3** | **4** | **>=5** |
| dmm\_flag | 0 | na | na | na | na | na |
| dmm\_mode | 0 | 0 | 0 | 0 | 0 | 0 |
| wedge\_full\_tab\_idx | 0 | 0 | 0 | 0 | 0 | 0 |
| dmm\_delta\_end\_flag | 0 | na | na | na | na | na |
| dmm\_delta\_end\_abs\_minus1 | 0 | 0 | 0 | 0 | 0 | 0 |
| dmm\_dc\_1\_abs dmm\_dc\_2\_abs | 0 | 0 | 0 | 0 | 0 | 0 |
| res\_pred\_flag | 0 | na | na | na | na | na |
| ic\_flag | 0 | na | na | na | na | na |
| mvp\_l0\_idx, mvp\_l1\_idx | 0 | 1 | na | na | na | na |
| residual\_dlt\_index\_flag | 0 | 0 | 0 | 0 | 0 | 0 |
| sdc\_flag | 0,1,2 (subclause G.9.3.3.1.1) | na | na | na | na | na |
| sdc\_residual\_flag | 0,1 (subclause G.9.3.3.1.7) | na | na | na | na | na |
| sdc\_residual\_sign\_flag | 0,1 (subclause G.9.3.3.1.7) | na | na | na | na | na |
| sdc\_residual\_abs\_minus1 | 0,10 (subclause H.9.3.3.1.7) | | | | | |
| sdc\_pred\_mode | 0 | 1 | 2 | na | na | na |
|  |  |  |  |  |  |  |