|  |  |
| --- | --- |
| **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**  2nd Meeting: Shanghai, CN, 13–19 Oct. 2012 | Document: JCT3V-B0050\_proposed text |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Proposed text for JCT3V-B0050 based on 3D-HEVC Test Model 1** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Li Zhang  Ying Chen Marta Karczewicz  5775 Morehouse Drive San Diego, CA 92121 USA | Tel: Email: | 1-858-651-6660 [lizhang@qti.qualcomm.com](mailto:lizhang@qti.qualcomm.com)  1-858-845-6589 [cheny@qti.qualcomm.com](mailto:cheny@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document provides text for Draft 1 of 3D-HEVC Test Model Description. The proposal only changes the motion vector candidate list derivation process as specified in sub-clause G.8.5.2.1.

The proposed specification text changes are as follows, with newly added/modified text highlighted in green and deleted text marked as red strikethrough ~~(red strikethrough)~~.

In the slice level, the value of the additional target reference index (refIdxLXA) for each of RefPicList0 and RefPicList1, and two flags bZeroIdxInterViewFlagLX if available, is derived.

**G.8.3.5 Derivation process for the additional target reference index for TMVP**

This process is invoked when the current slice is a P or B slice. Additional target reference indices refIdxL0A and refIdxL1A are derived which repsent different reference picture types from the reference picture with reference index equal to zero. In addtion, two flags bZeroIdxInterViewFlagL0 and bZeroIdxInterViewFlagL1 are derived to indicate whether the reference picture with index equal to 0 is an inter-view refernce picture.

Set variables refIdxL0A and refIdxL1A both to -1.

The following apply to derive refIdxL0A.

bZeroIdxInterViewFlagL0 = RefPicList0[ 0 ] is a short-term reference picture ? 0 : 1  
bFound = 0  
for( i = 1; i <= num\_ref\_idx\_l0\_active\_minus1&&! bFound; i++)  
 if ( (bZeroIdxInterViewFlagL0 && RefPicList0[ i ] is a short-term reference picture) | |  
 (!bZeroIdxInterViewFlagL0 && RefPicList0[ i ] is an inter-view reference picture) ) {  
 refIdxL0A = i  
 bFound =1  
 }

When the slice is a B slice, the following apply to derive refIdxL1A.

bZeroIdxInterViewFlagL1 = RefPicList1[ 0 ] is a short-term reference picture ? 0 : 1  
bFound = 0  
for( i = 1; i <= num\_ref\_idx\_l1\_active\_minus1&&! bFound; i++)  
 if ( (bZeroIdxInterViewFlagL1 && RefPicList1[ i ] is a short-term reference picture) | |  
 (!bZeroIdxInterViewFlagL1 && RefPicList1[ i ] is an inter-view reference picture) ) {  
 refIdxL1A = i  
 bFound =1  
 }

**G.8.5.2.1.1 Derivation process for luma motion vectors for merge mode**

…

The motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, and the prediction utilization flags predFlagL0 and predFlagL1 are derived as specified by the following ordered steps:

1. The derivation process for merging candidates from neighboring prediction unit partitions in subclause 8.5.2.1.2 is invoked with the luma coding block location ( xC, yC ), the coding block size nCS, the luma prediction block location ( xP, yP ), the variable singleMCLFlag, the width and the height of the luma prediction block nPbW and nPbH and the partition index partIdx as inputs and the output is assigned to the availability flags availableFlagN, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N and the motion vectors mvL0N and mvL1N with N being replaced by A0, A1, B0, B1 or B2.
2. The reference index for temporal merging candidate refIdxLX (with X being 0 or 1) is set equal to 0.
3. The derivation process for temporal luma motion vector prediction in subclause 8.5.3.1.7 is invoked with luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, ~~and~~ refIdxLX and mergeTMVP equal to 1 as the inputs and with the output being the availability flag availableFlagLXCol and the temporal motion vector mvLXCol. The variables availableFlagCol and predFlagLXCol (with X being 0 or 1, respectively) are derived as specified below.
4. **…**

**…**

**G.8.5.3.1.5 Derivation process for luma motion vector prediction**

* …

The motion vector predictor mvpLX is derived in the following ordered steps.

1. The derivation process for motion vector predictor candidates from neighboring prediction unit partitions in subclause 8.5.3.1.6 is invoked with the luma coding block location ( xC, yC ), the coding block size nCS, the luma prediction block location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, refIdxLX (with X being 0 or 1, respectively), and the partition index partIdx as inputs and the availability flags availableFlagLXN and the motion vectors mvLXN with N being replaced by A, B as the output.
2. If both availableFlagLXA and availableFlagLXB are equal to 1 and mvLXA is not equal to mvLXB, availableFlagLXCol is set equal to 0, otherwise, the derivation process for temporal luma motion vector prediction in subclause 8.5.3.1.7 is invoked with luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, ~~and~~ refIdxLX (with X being 0 or 1, respectively) and mergeTMVP equal to 0 as the inputs and with the output being the availability flag availableFlagLXCol and the temporal motion vector predictor mvLXCol.

**…**

### Changes for TMVP

**8.5.3.1.7 Derivation process for temporal luma motion vector prediction**

Inputs to this process are

* a luma location ( xP, yP ) specifying the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture,
* variables specifying the width and the height of the luma prediction block, nPbW and nPbH,
* a reference index refIdxLX (with X being 0 or 1)~~.~~,
* a variable mergeTMVP.

Outputs of this process are

* the motion vector prediction mvLXCol,
* the availability flag availableFlagLXCol.

The function RefPicOrderCnt( picX, refIdx, LX ) returns the picture order count PicOrderCntVal of the reference picture with index refIdx from reference picture list LX of the picture picX and is specified as follows.

RefPicOrderCnt( picX, refIdx, LX ) = PicOrderCnt(RefPicListX[ refIdx ] of the picture picX) (8 141)

Depending on the values of slice\_type, collocated\_from\_l0\_flag, and collocated\_ref\_idx, the variable colPic, specifying the picture that contains the collocated partition, is derived as follows.

* If slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 0, the variable colPic specifies the picture that contains the collocated partition as specified by RefPicList1[ collocated\_ref\_idx ].
* Otherwise (slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 1 or slice\_type is equal to P), the variable colPic specifies the picture that contains the collocated partition as specified by RefPicList0[ collocated\_ref\_idx ].

Variable colPb and its position ( xPCol, yPCol ) are derived in the following ordered steps:

1. The variable colPb is derived as follows

yPRb = yP + nPbH (8‑151)

* + If ( yP >> Log2CtbSizeY ) is equal to ( yPRb >> Log2CtbSizeY ), the horizontal component of the right-bottom luma location of the current luma prediction block is defined by

xPRb = xP + nPbW (8‑152)

and the variable colPb is set as the luma prediction block covering the modified location given by ( ( xPRb >> 4 ) << 4, ( yPRb >> 4 ) << 4 ) inside the colPic.

* + Otherwise ( ( yP >> Log2CtbSizeY ) is not equal to ( yPRb >> Log2CtbSizeY ) ), colPb is marked as "unavailable".

1. When colPb is coded in an intra prediction mode or colPb is marked as "unavailable", the following applies.
   * Central luma location of the current prediction block is defined by

xPCtr = ( xP + ( nPbW >> 1 ) (8‑153)

yPCtr = ( yP + ( nPbH >> 1 ) (8‑154)

* + The variable colPb is set as the luma prediction block covering the modified location given by ( ( xPCtr >> 4 ) << 4, ( yPCtr >> 4 ) << 4 ) inside the colPic.

1. ( xPCol, yPCol ) is set equal to the top-left sample of the colPb relative to the top-left luma sample of the colPic.

If mergeTMVP is equal to 1 and LongTermRefPic( currPic, refIdxLX, ListX ) is not equal to LongTermRefPic( colPic, refIdxCol, listCol ), the following apply:

* If LongTermRefPic( colPic, refIdxCol, listCol ) is equal to 1, refIdxLX is set to be refIdxLXA when bZeroIdxInterViewFlagL0 is equal 0 and refIdxLXA is larger than 0, refIdxLX is set to 0 if bZeroIdxInterViewFlagL0 is euqal to 1.
* Otherwise, (LongTermRefPic( colPic, refIdxCol, listCol ) is equal to 0), refIdxLX is set to be refIdxLXA when bZeroIdxInterViewFlagL0 is equal 1 and refIdxLXA is larger than 0, refIdxLX is set to 0 if bZeroIdxInterViewFlagL0 is euqal to 0.

The variables mvLXCol and availableFlagLXCol are derived as follows.

* If one or more of the following conditions are true, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
  + colPb is coded in an intra prediction mode.
  + colPb is marked as "unavailable".
  + slice\_temporal\_mvp\_enable\_flag is equal to 0.
  + LongTermRefPic( currPic, refIdxLX, ListX ) is not equal to LongTermRefPic( colPic, refIdxCol, listCol ).
* Otherwise, the motion vector mvCol, the reference index refIdxCol, and the reference list identifier listCol are derived as follows.
  + If PredFlagL0[ xPCol ][ yPCol ] is equal to 0, mvCol, refIdxCol, and listCol are set equal to MvL1[ xPCol ][ yPCol ], RefIdxL1[ xPCol ][ yPCol ], and L1, respectively.
  + Otherwise (PredFlagL0[ xPCol ][ yPCol ] is equal to 1), the following applies.
  + If PredFlagL1[ xPCol ][ yPCol ] is equal to 0, mvCol, refIdxCol, and listCol are set equal to MvL0[ xPCol ][ yPCol ], RefIdxL0[ xPCol ][ yPCol ], and L0, respectively.
  + Otherwise (PredFlagL1[ xPCol ][ yPCol ] is equal to 1), the following assignments are made.
    - * If PicOrderCnt( pic ) of every picture pic in every reference picture lists is less than or equal to PicOrderCntVal, mvCol, refIdxCol, and listCol are set equal to MvLX[ xPCol ][ yPCol ], RefIdxLX[ xPCol ][ yPCol ] and LX, respectively with X being the value of X this process is invoked for.
      * Otherwise (PicOrderCnt( pic ) of at least one picture pic in at least one reference picture list is greater than PicOrderCntVal, mvCol, refIdxCol and listCol are set equal to MvLN[ xPCol ][ yPCol ], RefIdxLN[ xPCol ][ yPCol ] and LN, respectively with N being the value of collocated\_from\_l0\_flag.

and the variable availableFlagLXCol is set equal to 1 and the following applies.

* + If RefPicListX[ refIdxLX ] is a long-term reference picture, or PicOrderCnt( colPic ) – RefPicOrderCnt( colPic, refIdxCol, listCol ) is equal to PicOrderCntVal – PicOrderCnt( RefPicListX[ refIdxLX ] ),

mvLXCol = mvCol (8‑155)

* + Otherwise, mvLXCol is derived as scaled version of the motion vector mvCol as specified below

tx = ( 16384 + ( Abs( td ) >>1 ) ) / td (8‑156)

distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 ) >> 6 ) (8‑157)

mvLXCol =  Clip3( −32768, 32767, Sign2( distScaleFactor \* mvCol ) \*    
 ( (Abs( distScaleFactor \* mvCol ) + 127 ) >> 8 ) ) (8‑158)

where td and tb are derived as

td = Clip3( −128, 127, PicOrderCnt( colPic ) – RefPicOrderCnt( colPic, refIdxCol, listCol ) ) (8‑159)

tb = Clip3( −128, 127, PicOrderCntVal – PicOrderCnt( RefPicListX [ refIdxLX ] ) ) (8‑160)