|  |  |
| --- | --- |
| **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-B0048\_proposed text |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Proposed text for JCT3V-B0048 based on 3D-HEVC Test Model 1** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Li Zhang  Ying Chen Liu He  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.

All the changes are tracked.

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

This process is only invoked when PredMode[ xC ][ yC ] is equal to MODE\_SKIP or PredMode[ xC ][ yC ] is equal to MODE\_INTER and merge\_flag [ xP ][ yP ] is equal to 1, where ( xP, yP ) specify the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture.

Inputs of this process are

* a luma location ( xC, yC ) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
* a luma location ( xP, yP ) of the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture,
* a variable nCS specifying the size of the current luma coding block,
* variables specifying the width and the height of the luma prediction block, nPbW and nPbH,
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are

– the luma motion vectors mvL0 and mvL1,

– the reference indices refIdxL0 and refIdxL1,

– the prediction list utilization flags predFlagL0 and predFlagL1.

The variables singleMCLFlag is derived as follows.

* If log2\_parallel\_merge\_level\_minus2 is greater than 0 and nCS is equal to 8, singleMCLFlag is set to 1.
* Otherwise, singleMCLFlag is set to 0.

When singleMCLFlag is equal to 1, xP is set equal to xC, yP is set equal to yC, and both nPbW and nPbH are set equal to nCS.

NOTE – When singleMCLFlag is equal to 1, all the prediction units of the current coding unit share a single merge candidate list, which is identical to the merge candidate list of the 2Nx2N prediction unit.

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 spatial merge candidates as specified 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. [Ed. Note (CY): this aspect has been changed in the software yet.]
3. The derivation process for temporal luma motion vector prediction in subclause 8.5.2.1.7 is invoked with luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, and refIdxLX 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.

availableFlagCol = availableFlagL0Col | | availableFlagL1Col (G‑73)   
predFlagLXCol = availableFlagLXCol (G‑74)

1. Depending on multi\_view\_mv\_pred\_flag, the following applies.

– If multi\_view\_mv\_pred\_flag is equal to 0, the variable availableFlagInterView is set equal to 0.

– Otherwise (multi\_view\_mv\_pred\_flag is equal to 1), the derivation process for the inter-view merge candidate as specified in subclause G.8.5.2.1.9 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, and the reference view identifier refViewIdx set equal to 0 as the inputs and the outputs are the flag availableFlagInterViewList[4], the prediction utilization flags predFlagL0InterViewList[4] and predFlagL1InterViewList[4], the reference indices refIdxL0InterViewList[4] and refIdxL1InterViewList[4], and the motion vectors mvL0InterViewList[4] and mvL1InterViewList[4].

1. The merge candidate list, mergeCandList, is constructed as specified by the following ordered steps:.
   1. The variable numMergeCand is set equal to 0.
   2. When availableFlagInterViewList[0] is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to InterViewList[0] and the variable numMergeCand is increased by 1.
   3. When availableFlagInterViewList[1] is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to InterViewList[1] and the variable numMergeCand is increased by 1.
   4. When availableFlagA1 is equal to 1 and A1 is different from InterViewList[i] if availableFlagInterViewList[i] is equal to 1 (i is being 0 or 1),, the entry mergeCandList[ numMergeCand ] is set equal to A1 and the variable numMergeCand is increased by 1.
   5. When availableFlagB1 is equal to 1 and B1 is different from InterViewList[i] if availableFlagInterViewList[i] is equal to 1 (i is being 0 or 1), the entry mergeCandList[ numMergeCand ] is set equal to B1 and the variable numMergeCand is increased by 1.
   6. When availableFlagB0 is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to B0 and the variable numMergeCand is increased by 1.
   7. When availableFlagA0 is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to A0 and the variable numMergeCand is increased by 1.
   8. When availableFlagB2 is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to B2 and the variable numMergeCand is increased by 1.
   9. When availableFlagInterViewList[2] is equal to 1 and InterViewList[2] is different from InterViewList[i] if availableFlagInterViewList[i] is equal to 1 (i is being 0 or 1) and A1, B1, the entry mergeCandList[ numMergeCand ] is set equal to InterViewList[2] and the variable numMergeCand is increased by 1.
   10. When availableFlagInterView[3] is equal to 1 and InterViewList[3] is different from InterViewList[i] if availableFlagInterViewList[i] is equal to 1 (i is being 0 ,1 or 2) and A1, B1, the entry mergeCandList[ numMergeCand ] is set equal to InterViewList[3] and the variable numMergeCand is increased by 1.
   11. When availableFlagCol is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to Col and the variable numMergeCand is increased by 1.

G.8.5.2.1.9 Derivation process for the inter-view merge candidate

This process is not invoked when multi\_view\_mv\_pred\_flag is equal to 0.

Inputs to this process are:

– a luma location ( xP, yP ) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,

– variables nPSW and nPSH specifying the width and the height, respectively, of the current prediction unit,

– a view identifier refViewIdx specifying a reference view.

Outputs of this process are:

– a flag availableFlagInterViewList[4] specifying whether each of the inter-view merge candidate is available,

– the prediction utilization flags predFlagL0InterViewList[4] and predFlagL1InterViewList[4],

– the reference indices refIdxL0InterViewList[4] and refIdxL1InterViewList[4] (if availableFlagInterViewList[4] is equal to 1),

– the motion vectors mvL0InterViewList[4] and mvL1InterViewList[4] (if availableFlagInterView is equal to 1).

For each variable i from 0 to 3, the following applies:

The variable availableFlagInterViewList[i] is set equal to 0. The variables predFlagL0InterViewList[i] and predFlagL1InterViewList[i] are set equal to 0. The variables refIdxL0InterViewList[i] and refIdxL1InterViewList[i] are set equal to −1. Both components of the motion vectors mvL0InterView and mvL1InterView are set equal to 0. The variables IvpMvFlagL0[ xP, yP ] and IvpMvFlagL1[ xP, yP ] are set equal to 0.

The inter-view merge candidates are derived by the following ordered steps.

1. If i is equal to 0, the derivation process for a disparity vector as specified in subclause G.8.5.2.1.13 is invoked with the luma location ( xP, yP ), and the variables nPSW and nPSH, as the inputs and a flag availableDV and a disparity vector mvDisp and as the outputs.
2. Set offset equal to 0, –nPSWleft and nPSW if i is equal to 0, 2 and 3, respectively.
3. The derivation process for a temporal inter-view motion vector merging candidate as specified in subclause G.8.5.2.1.15 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, the prediction list indication X equal to 0, the view identifier refViewIdx, a flag availableDV and a disparity vector (mvDisp[0] + offset, mvDisp[1]) as the inputs and the outputs are the flag availableFlagL0InterViewList[i],the motion vector mvL0InterViewList[i], the reference index tmpRefIdxL0List[i] and shifted offset nPSWleft.
4. When the current slice is a B slices, the derivation process for the temporal inter-view motion vector merging candidate as specified in subclause G.8.5.2.1.15 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, prediction list indication X equal to 1, the view identifier refViewIdx, a flag availableDV and a disparity vector (mvDisp[0] + offset, mvDisp[1]) as the inputs and the outputs are the flag availableFlagL1InterViewList[i] and the motion vector mvL1InterView List[i] and the reference index tempRefIdxL1List[i].
5. If i is equal to 0, i is increased by 1. When i is equal to 1 or both availableFlagL0InterViewList[i] and availableFlagL1InterViewList[i] are equal to 0, disparity inter-view motion vector merging candidates are derived by the following ordered steps.
6. The derivation process for the disparity inter-view motion vector merging candidate as specified in subclause G.8.5.2.1.17 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, and prediction list indication X equal to 0, a flag availableDV and a disparity vector (mvDisp[0] + offset, mvDisp[1]) as the inputs and the outputs are the flag availableFlagL0InterViewList[i] and the motion vector mvL0InterViewList[i].
7. When the current slice is a B slices, the derivation process for disparity inter-view motion vector merging candidate as specified in subclause G.8.5.2.1.17 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, prediction list indication X equal to 1, a flag availableDV and a disparity vector (mvDisp[0] + offset, mvDisp[1]) as the inputs and the outputs are the flag availableFlagL1InterViewList[i] and the motion vector mvL1InterViewList[i].

The prediction utilization flags predFlagL0InterViewList[i] and predFlagL1InterViewList[i] and the variable availableFlagInterViewList[i] are derived by: (something missing)

**G.8.5.2.1.15** **Derivation process for a temporal inter-view motion vector merging candidate**

This process is not invoked when multi\_view\_mv\_pred\_flag is equal to 0.

Inputs to this process are:

– a luma location ( xP, yP ) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,

– variables nPSW and nPSH specifying the width and the height, respectively, of the current prediction unit,

– a prediction list indication X.

– a flag availableDV

– a disparity vector mvDisp.

Outputs of this process are:

– a flag availableFlagLXInterView specifying whether the inter-view motion vector candidate is available,

– a motion vector candidate mvLXInterView (if availableFlagLXInterView is equal to 1).

– a reference index refIdxLX (with X being equal to 0 or 1) specifying a reference picture in the reference picture list RefPicListLX,

– the width nPSWoffset

~~The derivation process for a disparity vector as specified in subclause G.8.5.2.1.13 is invoked with the luma location ( xP, yP ), and the variables nPSW and nPSH, as the inputs and a flag availableDV and a disparity vector mvDisp and as the outputs.~~

The reference layer luma location ( xRef, yRef ) is derived by

xRef = Clip3( 0, PicWidthInSamplesL – 1, xP + ( ( nPSW – 1 ) >> 1 ) + ( ( mvDisp[0] + 2 ) >> 2 ) ) (G‑104)  
yRef = Clip3( 0, PicHeightInSamplesL – 1, yP + ( ( nPSH – 1 ) >> 1 ) + ( ( mvDisp[1] + 2 ) >> 2 )) (G‑105)

Let refCU be the coding unit that covers the luma location ( xRef, yRef ) in the view component with ViewIdx equal to refViewIdx.

nPSWoffset is set to the width of left prediction unit of refCU if it is available, otherwise, it is set equal to 0.

…

**G.8.5.2.1.17 Derivation process for a disparity inter-view motion vector merging candidate**

This process is not invoked when multi\_view\_mv\_pred\_flag is equal to 0.

Inputs to this process are:

– a luma location ( xP, yP ) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,

– variables nPSW and nPSH specifying the width and the height, respectively, of the current prediction unit,

– a prediction list indication X,

– a view identifier refViewIdx specifying a reference view.

– a flag availableDV

– a disparity vector mvDisp.

Outputs of this process are:

– a flag availableFlagLXInterView specifying whether the inter-view motion vector candidate is available,

– a motion vector candidate mvLXInterView (if availableFlagLXInterView is equal to 1),

– a reference index refIdxLX (with X being equal to 0 or 1) specifying a reference picture in the reference picture list RefPicListLX.

[GT (Ed.) This derivation process is specified as done in software. However, it might be changed to use refViewIdx to restrict the reference view to be the base view or the view mvDisp is related to. ]

The flag availableFlagLXInterView is set equal to 0.

The following apply for each i from 0 to num\_ref\_idx\_lX\_active\_minus1, inclusive

* + When PicOrderCntVal of the picture RefPicListX[ i ] is equal to the PicOrderCntVal or the current picture and availableFlagLXInterView is equal to 0 the availableFlagLXInterView is set equal to 1 and the following ordered steps apply.
    1. ~~The derivation process for a disparity vector as specified in subclause G.8.5.2.1.13 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, as the inputs and a flag availableDV and a disparity vector mvDisp as the outputs.~~
    2. The motion vector mvLXInterView and the reference index RefPicListLX are derived by

mvLXInterView[ 0 ] = mvDisp[ 0 ] (G‑)  
mvLXInterView[ 1 ] = 0 (G‑)  
refIdxLX = i (G‑)