|  |  |
| --- | --- |
| **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**  3rd Meeting: Geneva, CH, 17–23 Jan. 2013 | Document: JCT3V-C0051\_proposed text |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Proposed text for JCT3V-C0051 based on 3D-HEVC Test Model 2** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Li Zhang  Ying Chen Liu He 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 2 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 highlighted in green.

G.8.5.2.1.10 Derivation process for the inter-view motion vector predictor candidate

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

Inputs to 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 luma sample of the current prediction unit relative to the top-left luma sample of the current picture,
* a variable nCS specifying the size of the current luma coding block,
* variables nPSW and nPSH specifying the width and the height, respectively, of the current prediction unit,
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.
* a reference index refIdxLX (with X being equal to 0 or 1) specifying a reference picture in the reference picture list RefPicListLX.

Outputs of this process are:

– a view order index refViewIdx specifying a reference view,

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

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

The derivation process for a disparity vector as specified in subclause G.8.5.2.1.13 is invoked with the luma locations ( xC, yC ) and ( xP, yP ), the coding block size nCS, the variables nPSW and nPSH, and the partition index partIdx as the inputs and the view order index refViewIdx, a flag availableDV and a disparity vector mvDisp as the outputs.

The variable refPicViewIdx is set equal to the variable ViewIdx of the RefPicListLX[ refIdxLX ].

[Ed. (GT): refViewIdx might be used here to further restrict the reference views]

Depending on the value of refPicViewIdx, the following applies:

* + If refPicViewIdx is not equal to ViewIdx, the derivation process for a disparity inter-view motion vector predictor candidate as specified in subclause G.8.5.2.1.17 ~~G.8.5.2.1.11~~ is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, ~~and~~ the disparity vector mvDisp, bMRG flag equal to 0 as the inputs and the outputs are the flag availableFlagLXInterView and the motion vector candidate mvLXInterView.
  + Otherwise (refPicViewIdx is equal to ViewIdx), the derivation process for a temporal inter-view motion vector candidate as specified in subclause G.8.5.2.1.15~~G.8.5.2.1.12~~ is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, the reference index refIdxLX, the disparity vector mvDisp, ~~and~~ the reference view identifier refViewIdx, bMRG flag equal to 0 as the inputs and the outputs are the flag availableFlagLXInterView and the motion vector candidate mvLXInterView.

[Ed. Note (CY): consider adding definition for inter-view disparity motion vector and disparity vector.]

~~G.8.5.2.1.11 Derivation process for a disparity inter-view motion vector predictor 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 disparity vector mvDisp specifying the disparity to the reference block.~~

~~Outputs of this process are:~~

* ~~a flag availableFlagLXInterView specifying whether the inter-view motion vector candidate is available,~~
* ~~a motion vector candidate mvLXInterView.~~

~~The flag availableFlagLXInterView is set equal to 1 and the motion vector candidate mvLXInterView is derived by~~

* 1. ~~mvLXInterView[ 0 ] = mvDisp[ 0 ] (G‑107)  
     mvLXInterView[ 1 ] = 0 (G‑108)~~

~~G.8.5.2.1.12 Derivation process for a temporal inter-view motion vector predictor 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 reference index refIdxLX (with X being equal to 0 or 1) specifying a reference picture in the reference picture list RefPicListLX,~~
* ~~a disparity vector mvDisp specifying the disparity to the reference block,~~
* ~~a view index refViewIdx specifying a reference view.~~

~~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).~~

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

* 1. ~~xRef = Clip3( 0, PicWidthInSamples~~~~L~~~~– 1, xP + ( ( nPSW – 1 ) >> 1 ) + ( ( mvDisp[ 0 ] + 2 ) >> 2 ) ) (G‑109)  
     yRef = Clip3( 0, PicHeightInSamples~~~~L~~~~– 1, yP + ( ( nPSH – 1 ) >> 1 ) + ( ( mvDisp[ 1 ] + 2 ) >> 2 ) ) (G‑110)~~

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

~~[Ed. (GT): In software pictures of already coded view components of the same time instant are searched in coding order to find a coding unit refCU that covers the luma location ( xRef, yRef ) and has a PredMode equal to MODE\_SKIP or MODE\_INTER and provides a valid temporal inter-view motion vector candidate. Moreover, the view with ViewIdx equal to 1 is skipped in this search if the current ViewIdx of the current view component is larger than 1 to avoid prediction from non-base views. This only works correctly with current CTC and should be fixed. ].~~

~~The flag availableFlagLXInterView is set equal to 0, and both components of mvLXInterView are set equal to 0.~~

~~When the variable PredMode for the coding unit refCU is equal to MODE\_SKIP or MODE\_INTER, the following ordered steps apply.~~

* 1. ~~For Y being replaced by 0 and 1, the following applies.~~ 
     + ~~The variable refPredFlagLY is set equal to the variable predFlagLY of the prediction unit refPU.~~
     + ~~The variable refMvLY is set equal to mvLY of the prediction unit refPU.~~
     + ~~The variable refRefPicListLY is set equal to the reference picture list RefPicListLY of refPU.~~
  2. ~~When refPredFlagL0 is equal to 1 and the picture order count of the picture refRefPicListL0[ refRefIdxL0 ] is equal to the picture order count of the picture RefPicListLX[ refIdxLX ], the flag availableFlagLXInterView is set equal to 1 and the motion vector mvLXInterView is derived by~~
     + 1. ~~mvLXInterView[ 0 ] = refMvL0[ 0 ] (G‑111)  
          mvLXInterView[ 1 ] = refMvL0[ 1 ] (G‑112)~~
  3. ~~When availableFlagLXInterView is equal to 0 and refPredFlagL1 is equal to 1 and the picture order count of the picture refRefPicListL1[ refRefIdxL1 ] is equal to the picture order count of the picture RefPicListLX[ refIdxLX ], the flag availableFlagLXInterView is set equal to 1 and the motion vector mvLXInterView is derived by~~
     + 1. ~~mvLXInterView[ 0 ] = refMvL1[ 0 ] (G‑113)  
          mvLXInterView[ 1 ] = refMvL1[ 1 ] (G‑114)~~

G.8.5.2.1.15 Derivation process for a temporal inter-view motion vector ~~merging~~ candidate for AMVP and Merge

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 disparity vector mvDisp,
* a prediction list indication X,
* a reference view index refViewIdx.
* a flag bMRG specifying whether current mode is merge, when not presented, it is inferred to be 1.
* a reference index refIdxLX specifying a reference picture in the reference picture list RefPicListLX, when not presented, it is inferred to be 0.

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 specifying a reference picture in the reference picture list RefPicListLX,

The flag availableFlagLXInterView is set equal to 0, both components of mvLXInterView are set equal to 0, and refIdxLX is set equal to −1.

When X is equal to 1 and the current slice is not a B slice the whole decoding process specified in this subclause terminates.

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

* 1. xRef = Clip3( 0, PicWidthInSamplesL – 1, xP + ( ( nPSW – 1 ) >> 1 ) + ( ( mvDisp[ 0 ] + 2 ) >> 2 ) ) (G‑127)  
     yRef = Clip3( 0, PicHeightInSamplesL – 1, yP + ( ( nPSH – 1 ) >> 1 ) + ( ( mvDisp[ 1 ] + 2 ) >> 2 )) (G‑128)

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

[Ed. (GT:) In software refCU is the coding unit that covers the luma location ( xRef, yRef ) in the view component that closest (in terms of view\_id) to the current view component and that is included in one of the reference lists of the current view component.]

[Ed. (CY): What is implemented in the software has no impact on the final results in common test condition, so it is preferred to have the current text and software changes may be desirable. ]

When the variable PredMode for the coding unit refCU is equal to MODE\_SKIP or MODE\_INTER, the following ordered steps apply:

* 1. The variable refPredFlagLX is set equal to the prediction utilization flag predFlagLX of the prediction unit refPU.
  2. The variable refRefIdxLX, is set equal to the reference index refIdxLX of the prediction unit refPU.
  3. The variable refMvLX is set equal to the motion vector mvLX of the prediction unit refPU.
  4. The variable refRefPicListLX, is set equal to the reference picture list RefPicListLX of the view component with ViewIdx equal to refViewIdx.
  5. When refPredFlagLX is equal to 1, the following applies for each i from (bMRG? 0 : refIdxLX) to (bMRG? num\_ref\_idx\_lX\_active\_minus1: refIdxLX), inclusive
     + When availableFlagLXInterView is equal to 0, and the picture order count of the picture refRefPicListLX[ refRefIdxLX ] is equal to the picture order count of the picture RefPicListLX[ i ], the flag availableFlagLXInterView is set equal to 1 and the following applies.
       - 1. mvLXInterView[ 0 ] = refMvLX[ 0 ] (G‑129)  
            mvLXInterView[ 1 ] = refMvLX[ 1 ] (G‑130)  
            if bMRG is equal to 1, refIdxLX = i (G‑131)
  6. Let refPredFlagLY, with Y being replaced by 0 and 1, be the variables predFlagLY for the prediction unit refPU. Let refRefIdxLY, with Y being replaced by 0 and 1, be the variables refIdxLY for the prediction unit refPU. Let refMvLY, with Y being replaced by 0 and 1, be the variables mvLY for the prediction unit refPU. Let refRefPicListLY, with Y being replaced by 0 and 1, be the reference picture list RefPicListLY for the prediction unit refPU in the view component with ViewIdx equal to refViewIdx.
  7. When refPredFlagLY is equal to 1 (with Y equal to 1-X), the following apply for each i from (bMRG? 0 refIdxLX) to (bMRG? num\_ref\_idx\_lX\_active\_minus1: refIdxLX), inclusive,
     + When availableFlagLXInterView is 0, and the picture order count of the picture refRefPicListLY[ refRefIdxLY ] is equal to the picture order count of the picture RefPicListLX[ i ], the flag availableFlagLXInterView is set equal to 1 and the following applies.

mvLXInterView[ 0 ] = refMvLY[ 0 ] (G‑xxx)  
mvLXInterView[ 1 ] = refMvLY[ 1 ] (G‑xxx)  
if bMRG is equal to 1, refIdxLX = i (G‑xxx)

G.8.5.2.1.17 Derivation process for a disparity inter-view motion vector ~~merging~~ candidate for merge and AMVP

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 bMRG specifying whether current mode is merge, when not presented, it is inferred to be 1.

Outputs of this process are:

* a view order index refViewIdx specifying a reference view.
* 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, both components of mvLXInterView are set equal to 0, and refIdxLX is set equal to −1.

When X is equal to 1 and the current slice is not a B slice the whole decoding process specified in this subclause terminates.

The following apply for each i from (bMRG ? 0: refIdxLX) to (bMRG ? num\_ref\_idx\_lX\_active\_minus1: refIdxLX), inclusive

* + When PicOrderCntVal of the picture RefPicListX[ i ] is equal to the PicOrderCntVal or the current picture and availableFlagLXInterView is equal to 0, availableFlagLXInterView is set equal to 1 and the motion vector mvLXInterView and the reference index RefPicListLX are derived by
    - 1. mvLXInterView[ 0 ] = mvDisp[ 0 ] (G‑136)  
         mvLXInterView[ 1 ] = 0 (G‑137)  
         refIdxLX = i (G‑138)