The modifications are highlighted in Green.

H.8.2.1.10 Derivation process for a temporal inter-view motion vector candidate

This process is not invoked when iv\_mv\_pred\_flag[ nuh\_layer\_id ] 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 reference view index refViewIdx.
* a disparity vector mvDisp,
* a flag mergeFlag specifying whether a merge candidate is derived,
* a reference index refIdxLX specifying a reference picture in the reference picture list RefPicListLX.

Outputs of this process are:

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

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

xRef = Clip3( 0, PicWidthInSamplesL – 1, ((xP + ( ( nPSW – 1 ) >> 1 ) + ( ( mvDisp[ 0 ] + 2 ) >> 2 ))>>3) <<3 ) (H‑124)  
yRef = Clip3( 0, PicHeightInSamplesL – 1, ((yP + ( ( nPSH – 1 ) >> 1 ) + ( ( mvDisp[ 1 ] + 2 ) >> 2 ))>>3)<<3 ) (H‑125)

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 with smallest ViewIdx and that is included in one of the reference lists of the current view component.(#6)]

[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, for Y in the range of X to (1 – X), inclusive:

* 1. The variable refPredFlagLY is set equal to the prediction utilization flag predFlagLY of the prediction unit refPU.
  2. The variable refRefIdxLY, is set equal to the reference index refIdxLY of the prediction unit refPU.
  3. The variable refMvLY is set equal to the motion vector mvLY of the prediction unit refPU.
  4. The variable refRefPicListLY, is set equal to the reference picture list RefPicListLY of the prediction unit refCU.
  5. When refPredFlagLY is equal to 1, the following applies for each i from ( mergeFlag ? 0 : refIdxLX ) to ( mergeFlag ? num\_ref\_idx\_lX\_active\_minus1 : refIdxLX), inclusive:
     + When availableFlagLXInterView is equal to 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.
       - The motion vector mvLXInterView is derived by:
         1. mvLXInterView[ 0 ] = refMvLY[ 0 ] (H‑)  
            mvLXInterView[ 1 ] = refMvLY[ 1 ] (H‑)
       - When mergeFlag is equal to 1, the reference index refIdxLX is derived by:
         1. refIdxLX = i (H‑)

H.8.5.2.1.13 Derivation process for the texture merging candidate

This process is not invoked when DepthFlag 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,

Outputs of this process are:

* a flag availableFlagT specifying whether the texture merging candidate is available,
* the prediction utilization flags predFlagL0T and predFlagL1T,
* the reference indices refIdxL0T and refIdxL1T (when availableFlagT is equal to 1),
* the motion vectors mvL0T and mvL1T (when availableFlagT is equal to 1).

The variable availableFlagT is set equal to 0. The variables predFlagL0T and predFlagL1T are set equal to 0. The variables refIdxL0T and refIdxL1T are set equal to −1. Both components of the motion vectors mvL0T and mvL1T are set equal to 0.

The texture luma location ( xRef, yRef ) is derived by:

xRef = Clip3( 0, PicWidthInSamplesL – 1,((xP + ( ( nPSW – 1 ) >> 1 ))>>3)<<3 ) (H‑)

yRef = Clip3( 0, PicHeightInSamplesL – 1, ((yP + ( ( nPSH – 1 ) >> 1 ))>>3)<<3 ) (H‑)

* 1. [ Ed. (GT): Is clipping necessary? ]

Let textPic be the picture with PicOrderCntVal and ViewIdx equal to PicOrderCnt and ViewIdx of the current picture and DepthFlag being equal to 0 and let textPU the PU at covering the position ( xRef, yRef ) in textPic.

For X in the range of 0 to 1, inclusive, the following applies:

[Ed. (GT): It should be verified that the picture with refIdxLX is also a reference picture for depth. (#10)]

* 1. The variable textPredFlagLX is set equal to PredFlagLX of textPic. The variable textRefIdxLX is set equal to RefIdxLX of textPic. The variable textMvLX is set equal to the MvLX of textPic.
  2. When X is equal to 0 or the current slice is a B slice, the following applies:
     + When textPredFlagLX[ xRef ][ yRef ] is equal to 1, the following applies:
       1. mvLXT[ 0 ] = ( textMvLX[ xRef ][ yRef ][ 0 ] + 2 ) >> 2 (H‑)
       2. mvLXT[ 1 ] = ( textMvLX[ xRef ][ yRef ][ 1 ] + 2 ) >> 2 (H‑)
       3. refIdxLX = textRefIdxLX[ xRef ][ yRef ] (H‑)
       4. predFlagLXT = 1 (H‑)
       5. availableFlagT = 1 (H‑)