The modification is based on JCT3V-G1001\_v2, the added parts are highlighted with Green, and the deleted parts are marked as ~~strikethrough~~.

Derivation process for inter layer predicted sub prediction block motion vector candidates

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 ( xPb, yPb ) of the top-left sample of the current prediction luma block relative to the top-left luma sample of the current picture,
* two variables nPbW and nPbH specifying the width and the height of the current prediction block,
* a reference view index refViewIdx,
* a disparity vector mvDisp.

Outputs of this process are:

* the flags availableFlagLXN, with X in the range of 0 to 1, inclusive, specifying whether the inter-view or inter-component predicted merging candidate is available,
* the inter-view or inter-component predicted merging candidate mvLXN, with X in the range of 0 to 1, inclusive,
* the reference index refIdxLXN, with X in the range of 0 to 1, inclusive, specifying a reference picture in the reference picture list RefPicListLX.

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

– The flag availableFlagLXN is set equal to 0.

– The motion vector mvLXN is set equal to ( 0, 0 ).

– The reference index refIdxLXN is set equal to −1.

The variables minSize, nSbW and nSbH are derived as:

* 1. minSize = DepthFlag ? MpiSubPbSize : SubPbSize[ nuh\_layer\_id ] (I‑170)
  2. nSbW = ( nPbW / SubPbSize[ nuh\_layer\_id ] <= 1 ) ? nPbW : minSize (I‑171)
  3. nSbH = ( nPbH / SubPbSize[ nuh\_layer\_id ] <= 1 ) ? nPbH : minSize (I‑172)

Depending on DepthFlag, the following applies:

* If DepthFlag is equal to 0, for X in the range of 0 to 1, inclusive, the derivation process for a temporal inter-view motion vector candidate as specified in subclause I.8.5.3.2.11 is invoked with the luma location ( xPb + ( nPbW / nSbW / 2 ) \* nSbW, yPb + ( nPbH / nSbH / 2 ) \* nSbH ), the variables nSbW and nSbH, the prediction list indication X, the view order index refViewIdx, and the disparity vector mvDisp as inputs, and the outputs are the flag availableFlagLXN, the motion vector mvLXN and the reference index refIdxLXN.
* Otherwise (DepthFlag is equal to 1), the following applies:
  + ~~The variable lastAvailableFlag is set equal to 0.~~
  + ~~The variables xLastAvail and yLastAvail are both set equal to −1.~~
  + ~~The variable curSubBlockIdx is set equal to 0.~~
* If DepthFlag is equal to 1, for X in the range of 0 to 1, inclusive, the derivation process for a texture merging candidate as specified in subclause  is invoked with the luma location ( xPb + ( nPbW / nSbW / 2 ) \* nSbW, yPb + ( nPbH / nSbH / 2 ) \* nSbH ), the variables nSbW and nSbH, and the variable mvAccFlag being equal to 0 as inputs, and the outputs are the flag availableFlagLXN, the motion vector mvLXN and the reference index refIdxLXN.

[Ed. (GT): In original proposal there was an additional check, whether the slice in picture with refViewIdx is an I slice. This check seems to be redundant and has not been integrated. Please inform me, when this is not correct.]

When availableFlagL0InterView or availableFlagL1InterView is equal to 1, the following applies:

* For yBlk in the range of 0 to ( nPbH / nSbH − 1 ), inclusive, the following applies:
  + For xBlk in the range of 0 to ( nPbW / nSbW − 1 ), inclusive, the following applies:
    - If DepthFlag is equal to 0, the following applies:
      * For X in the range of 0 to 1, inclusive, the derivation process for a temporal inter-view motion vector candidate as specified in subclause I.8.5.3.2.11 is invoked with the luma location ( xPb + xBlk \* nSbW, yPb + yBlk \* nSbH ), the variables nSbW and nSbH, the prediction list indication X, the view order index refViewIdx, and the disparity vector mvDisp as inputs, and the outputs are the flag spPredFlagLX[ xBlk ][ yBlk ], the motion vector spMvLX[ xBlk ][ yBlk ] and the reference index spRefIdxLX[ xBlk ][ yBlk ].
      * When spRefIdxL0[ xBlk ][ yBlk ] and spRefIdxL1[ xBlk ][ yBlk ] are both equal to −1, the following applies for X in the range of 0 to 1, inclusive:

spMvLX[ xBlk ][ yBlk ] = mvLXN (I‑173)

spRefIdxLX[ xBlk ][ yBlk ] = refIdxLXN (I‑174)

spPredFlagLX[ xBlk ][ yBlk ] = availableFlagLXN (I‑175)

* + - Otherwise (DepthFlag is equal to 1), the following applies:
      * For X in the range of 0 to 1, inclusive, the derivation process for a texture merging candidate as specified in subclause I.8.5.3.2.14 is invoked with the luma location ( xPb + xBlk \* nSbW, yPb + yBlk \* nSbH ), the variables nSbW and nSbH, and the variable mvAccFlag being equal to 0 as inputs, and the outputs are the flag curAvailableFlag, spPredFlagLX[ xBlk ][ yBlk ], the reference index spRefIdxLX[ xBlk ][ yBlk ] and the motion vector spMvLX[ xBlk ][ yBlk ].
      * When curAvailableFlag is equal to -1, the following applies for X in range of 0 to 1, inclusive:
        + spMvLX[ xBlk ][ yBlk ] = mvLXN (‑173)
        + spRefIdxLX[ xBlk ][ yBlk ] = refIdxLXN (‑174)
        + spPredFlagLX[ xBlk ][ yBlk ] = availableFlagLXN (‑175)
      * ~~Depending on curAvailableFlag, the following applies:~~
        + ~~If curAvailableFlag is equal to 1, the following ordered steps apply:~~

~~When lastAvailableFlag is equal to 0, the following applies:~~

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

~~mvLXN = spMvLX[ xBlk ][ yBlk ] (‑176)~~

~~refIdxLXN = spRefIdxLX[ xBlk ][ yBlk ] (‑177)~~

~~predFlagLXN = spPredFlagLX[ xBlk ][ yBlk ] (‑178)~~

~~When curSubBlockIdx is greater than 0, the following applies for k in the range of 0 to ( curSubBlockIdx − 1 ), inclusive:~~

~~The variables i and k are derived as specified in the following:~~

~~i = k % ( nPbW / nSbW ) (‑179)~~

~~j = k / ( nPbW / nSbW ) (‑180)~~

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

~~spMvLX[ i ][ j ] = spMvLX[ xBlk ][ yBlk ] (‑181)~~

~~spRefIdxLX[ i ][ j ] = spRefIdxLX[ xBlk ][ yBlk ] (‑182)~~

~~spPredFlagLX[ i ][ j ] = spPredFlagLX[ xBlk ][ yBlk ] (‑183)~~

~~The variable lastAvailableFlag is set equal to 1.~~

~~The variables xLastAvail and yLastAvail are set equal to xBlk and yBlk, respectively.~~

* + - * + ~~Otherwise (curAvailableFlag is equal to 0), when lastAvailable Flag is equal to 1, the following applies for X in the range of 0 to 1, inclusive:~~

~~spMvLX[ xBlk ][ yBlk ] = spMvLX[ xLastAvail ][ yLastAvail ] (‑184)~~

~~spRefIdxLX[ xBlk ][ yBlk ] = spRefIdxLX[ xLastAvail ][ yLastAvail ] (‑185)~~

~~spPredFlagLX[ xBlk ][ yBlk ] = spPredFlagLX[ xLastAvail ][ yLastAvail ] (‑186)~~

* + - * ~~The variable curSubBlockIdx is set equal to curSubBlockIdx + 1.~~

[Ed. (CY/GT): The motion hole filling process in sub-PU MPI hasn’t been explicitly proposed to be harmonized with sub-PU inter-view motion prediction. The perspective harmonization can make major parts of the text of “Otherwise (DepthFlag is equal to 1), the following applies” branches unnecessary.]

* For use in derivation processes of variables invoked later in the decoding process, the following assignments are made for x = 0.. nPbW − 1 and y = 0.. nPbH − 1:
  + - The variable SubPbPartIdc is derived as specified in following:
      * 1. SubPbPartIdc[ xPb + x ][ yPb + y ] = SUB\_PART\_DEFAULT (‑187)
    - For X in the range of 0 to 1, inclusive, the following applies:
      * The variables SubPbPredFlagLX, SubPbMvLX and SubPbRefIdxLX are derived as specified in following:

SubPbPredFlagLX[ xPb + x ][ yPb + y ] = spPredFlagLX[ x / nSbW ][ y / nSbW ] (‑188)

SubPbMvLX[ xPb + x ][ yPb + y ] = spMvLX[ x / nSbW ][ y / nSbW ] (‑189)

SubPbRefIdxLX[ xPb + x ][ yPb + y ] = spRefIdxLX[ x / nSbW ][ y / nSbW ] (‑190)

* + - * The derivation process for chroma motion vectors in subclause 8.5.3.2.9 is invoked with SubPbMvLX[ xPb + x ][ yPb + y ] as input and the output is SubPbMvCLX[ xPb + x ][ yPb + y ].