1. * + - 1. Derivation process for a view synthesis prediction merge candidate

Inputs to this process are:

* a luma location ( xCb, yCb ) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,

Outputs of this process are

* the availability flag availableFlagVSP whether the VSP merge candidate is available,
* the reference indices refIdxL0VSP and refIdxL1VSP ,
* the prediction list utilization flags predFlagL0VSP and predFlagL1VSP,
* the motion vectors mvL0VSP and mvL1VSP.

1. The variable availableFlagVSP is set equal to 1, the variables predFlagL0VSP and predFlagL1VSP are set equal to 0, the variables refIdxL0VSP and refIdxL1VSP are set equal to −1 and the variable refViewAvailableFlag is set equal to 0.

* For X in the range of 0 to 1, inclusive, the following applies:
  + For i in the range of 0 to NumRefPicsLX – 1, inclusive, the following applies:
    - When refViewAvailableFlag is equal to 0 and ViewIdx( RefPicListX[ i ] ) is equal to RefViewIdx[ xCb ][ yCb ], the following applies:
      * 1. refViewAvailableFlag = 1 (‑133)
        2. predFlagLXVSP = 1 (‑134)
        3. mvLXVSP = MvDisp[ xCb ][ yCb ] (‑135)
        4. refIdxLXVSP = i (‑136)
        5. Y = 1 – X (‑137)
        7. Derivation process for a sub prediction block 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 ( xPb, yPb ) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,
* variables nPbW and nPbH specifying the width and the height, respectively, of the current prediction unit,
* a reference view index refViewIdx.
* a disparity vector mvDisp,

Outputs of this process are:

* the flags availableFlagLXInterView, with X in the range of 0 to 1, inclusive, specifying whether the temporal inter-view motion vector candidate is available,
* the temporal inter-view motion vector candidate mvLXInterView, with X in the range of 0 to 1, inclusive.
* the reference index refIdxLXInterView, 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 availableFlagLXInterView is set equal to 0.

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

– The reference index refIdxLXInterView is set equal to –1.

The variables nSbW and nSbH are derived as:

* 1. nSbW = nPbW / SubPbSize[ nuh\_layer\_id ] <= 1 ? nPbW : SubPbSize[ nuh\_layer\_id ] (‑157)
  2. nSbH = nPbH / SubPbSize[ nuh\_layer\_id ] <= 1 ? nPbH : SubPbSize[ nuh\_layer\_id ] (‑158)

The variable ivRefPic is set equal to the picture with ViewIdx equal to refViewIdx in the current access unit, the variable curSubBlockIdx is set equal to 0 and the variable lastAvailableFlag is set equal to 0.

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:
  + - 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 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 the inputs and the outputs are the flag spPredFlagLX[ xBlk ][ yBlk ], the motion vector spMvLX[ xBlk ][ yBlk ] and the reference index spRefIdxLX[ xBlk ][ yBlk ].
    - If all of the following conditions are true, the spPredFlagL1[xBlk][yBlk] is set equal to 0 and spRefIdxL1[xBlk][yBlk] is set equal to -1
      * nSbW\*nSbH < 64
      * spPredFlagL0[ xBlk ][ yBlk ] == 1
      * spPredFlagL1[ xBlk ][ yBlk ] == 1
  + The variable curAvailableFlag is set equal to ( spRefIdxL0[ xBlk ][ yBlk ]  | |  spRefIdxL1[ xBlk ][ yBlk ] ).
  + Depending on curAvailableFlag, the following applies:
    - If curAvailableFlag is equal to 1, the following ordered steps apply:
      1. When lastAvailableFlag is equal to 0, the following applies:
         * For X in the range of 0 to 1, inclusive, the following applies:

mvLXInterView = spMvLX[ xBlk ][ yBlk ] (‑159)

refIdxLXInterView = spRefIdxLX[ xBlk ][ yBlk ] (‑160)

availableFlagLXInterView = spPredFlagLX[ xBlk ][ yBlk ] (‑161)

* + - * + 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 ) (‑162)

j = k / ( nPbW / nSbW ) (‑163)

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

spMvLX[ i ][ j ] = spMvLX[ xBlk ][ yBlk ] (‑164)

spRefIdxLX[ i ][ j ] = spRefIdxLX[ xBlk ][ yBlk ] (‑165)

spPredFlagLX[ i ][ j ] = spPredFlagLX[ xBlk ][ yBlk ] (‑166)

* + - 1. The variable lastAvailableFlag is set equal to 1.
      2. 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:
      * 1. spMvLX[ xBlk ][ yBlk ] = spMvLX[ xLastAvail ][ yLastAvail ] (‑167)
        2. spRefIdxLX[ xBlk ][ yBlk ] = spRefIdxLX[ xLastAvail ][ yLastAvail ] (‑168)
        3. spPredFlagLX[ xBlk ][ yBlk ] = spPredFlagLX[ xLastAvail ][ yLastAvail ] (‑169)
  + The variable curSubBlockIdx is set equal to curSubBlockIdx + 1.

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:

* + 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:
      * 1. SubPbPredFlagLX[ xPb + x ][ yPb + y ] = spPredFlagLX[ x / nSbW ][ y / nSbW ] (‑170)
        2. SubPbMvLX[ xPb + x ][ yPb + y ] = spMvLX[ x / nSbW ][ y / nSbW ] (‑171)
        3. SubPbRefIdxLX[ xPb + x ][ yPb + y ] = spRefIdxLX[ x / nSbW ][ y / nSbW ] (‑172)
    - 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 ].