G.8.5.2.1.13 Derivation process for a disparity vector

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.

Outputs of this process are:

* a view order index refViewIdx specifying a reference view.
* a flag availableDV specifying whether the disparity vector is available,
* a disparity vector mvDisp.

The flag availableDV is set equal to 0, and both components of the disparity vector mvDisp are set equal to 0.

The variable checkParallelMergeFlag is derived as follows.

* If one or more of the following conditions are true, checkParallelMergeFlag is set equal to 1.
  + PredMode[ xC ][ yC ] is equal to MODE\_SKIP
  + PredMode[ xC ][ yC ] is equal to MODE\_INTER and merge\_flag[ xP ][ yP ] is equal to 1.
* Otherwise, checkParallelMergeFlag is set equal to 0.

The motion vector mvDisp and the availability flag availableDV are derived in the following ordered steps and the whole decoding process of this sub-clause terminates once availableDV is set equal to 1.

* 1. The derivation process for a disparity vector from temporal neighbour block as specified in G.8.5.2.1.16 is invoked with a luma location ( xP, yP ), variables nPSW and nPSH as inputs, and flag availableDV, and disparity vector mvDisp outputs

2. For each N being A1, B1, B0, A0, or B2, and ( xN, yN ) being ( xP − 1,  yP + nPSH − 1 ), ( xP + nPSW − 1,  yP − 1 ), ( xP + nPSW,  yP − 1 ), ( xP − 1,  yP + nPSH  ), or ( xP − 1,  yP − 1 ), the following applies.

1. When yP−1 is less than (( yC >> Log2CtbSizeY ) << Log2CtbSizeY), the following applies.
   * + - 1. xB0 = ((xB0>>3)<<3) + ((xB0>>3)&1)\*7 (G‑115)  
            xB1 = ((xB1>>3)<<3) + ((xB1>>3)&1)\*7 (G‑116)  
            xB2 = ((xB2>>3)<<3) + ((xB2>>3)&1)\*7 (G‑117)
2. The availability derivation process for a prediction block as specified in subclause 6.4.2 is invoked with the luma location ( xC, yC ), the current luma coding block size nCbS set equal to nCS, the luma location ( xP, yP ), the width and the height of the luma prediction block nPbW and nPbH, the partition index partIdx, and the luma location ( xN, yN ) as inputs and the output is assigned to the prediction block availability flag availableN.
3. When all of the following conditions are true, availableA1 is set equal to 0.
   * + - N is equal to A1
       - PartMode of the current prediction unit is PART\_Nx2N or PART\_nLx2N or PART\_nRx2N
       - partIdx is equal to 1
4. When all of the following conditions are true, availableB1 is set equal to 0.
   * + - N is equal to B1
       - PartMode of the current prediction unit is PART\_2NxN or PART\_2NxnU or PART\_2NxnD
       - partIdx is equal to 1
5. When all of the following conditions are true, availableN is set equal to 0.
   * + - checkParallelMergeFlag is equal to 1
       - (xP >> (log2\_parallel\_merge\_level\_minus2 + 2) ) is equal to  
         (xN >> (log2\_parallel\_merge\_level\_minus2 + 2) )
       - (yP >> (log2\_parallel\_merge\_level\_minus2 + 2) ) is equal to (yN >> (log2\_parallel\_merge\_level\_minus2 + 2) ).
6. For each X from 0 to 1, the following applies, when availableN is equal to 1 and PredFlagLX[ xN ][ yN ] is equal to 1
   * + - If RefPicListX[ RefIdxLX[ xN ][ yN ] ] is an inter-view reference picture, the following applies:

refViewIdx = ViewIdx( RefPicListX[ RefIdxLX[ xN ][ yN ] ] ) (G‑118)  
mvDisp = MvLXN[ xN ][ yN ] (G‑119)  
availableDV = 1 (G‑120)

(G‑122)