H.8.5.3.2.1 Derivation process for luma motion vectors for merge mode

This process is only invoked when merge\_flag[ xPb ][ yPb ] is equal to 1, where ( xPb, yPb ) specify the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture.

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,
* a luma location ( xPb, yPb ) of the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture,
* a variable nCbS specifying the size of the current luma coding block,
* two variables nPbW and nPbH specifying the width and the height of the luma prediction block,
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are:

* the luma motion vectors mvL0 and mvL1,
* the reference indices refIdxL0 and refIdxL1,
* the prediction list utilization flags predFlagL0 and predFlagL1,
* the disparity vector availability flags ivpMvFlagL0 and ivpMvFlagL1,
* the flag vspModeFlag, specifying, whether the current PU is coded using view synthesis prediction,
* the flag subPbMotionFlag, specifying, whether the motion data of the current PU has sub prediction block size motion accuracy.

The function differentMotion( N, M ) is specified as follows:

* If one of the following conditions is true, differentMotion( N, M ) is equal to 1:
  + predFlagLXN != predFlagLXM (with X being replaced by 0 and 1),
  + mvLXN != mvLXM (with X being replaced by 0 and 1),
  + refIdxLXN != refIdxLXM (with X being replaced by 0 and 1),
* Otherwise, differentMotion( N, M ) is equal to 0.

The motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, and the prediction utilization flags predFlagL0 and predFlagL1 are derived by the following ordered steps:

* 1. The derivation process for the base merge candidate list as specified in subclause H.8.5.3.2.18 is invoked with the luma location ( xCb, yCb ), the luma location ( xPb, yPb ), the variables nCbS, nPbW, nPbH, and the partition index partIdx as inputs, and the output being a modified luma location ( xPb, yPb ), the modified variables nPbW and nPbH, the modified variable partIdx, the luma location ( xOrigP, yOrigP ), the variables nOrigPbW and nOrigPbH, the merge candidate list baseMergeCandList, the luma motion vectors mvL0N and mvL1N, the reference indices refIdxL0N and refIdxL1N, and the prediction list utilization flags predFlagL0N and predFlagL1N, with N being replaced by all elements of baseMergeCandList.
  2. For N being replaced by A1, B1, B0, A0 and B2, the following applies:
     + If N is an element in baseMergeCandList, availableFlagN is set equal to 1, the derivation process for a view synthesis prediction flag as specified in subclause H.8.5.3.2.17 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, the merge candidate indicator N as the inputs, and the output is the mergeCandIsVspFlagN.
     + Otherwise (N is not an element in baseMergeCandList), availableFlagN is set equal to 0, mergeCandIsVspFlagN is set equal to 0.
  3. For N being replaced by B1 and B0, the following applies:
     + If N is equal to B1, mergeCandIsVspFlagB1 is set equal to (!mergeCandIsVspFlagA1 && mergeCandIsVspFlagB1).
     + If N is equal to B0, mergeCandIsVspFlagB0 is set equal to (!(mergeCandIsVspFlagA1 || mergeCandIsVspFlagB1) && mergeCandIsVspFlagB0).
  4. Depending on iv\_mv\_pred\_flag[ nuh\_layer\_id ], the following applies:
     + If iv\_mv\_pred\_flag[ nuh\_layer\_id ] is equal to 0, the flags availableFlagIvMC, availableIvMCShift and availableFlagIvDC are set equal to 0.
     + Otherwise (iv\_mv\_pred\_flag[ nuh\_layer\_id ] is equal to 1), the derivation process for the inter-view merge candidates as specified in subclause H.8.5.3.2.10 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, as the inputs and the output is assigned to the availability flags availableFlagIvMC, availableIvMCShift and availableFlagIvDC, the reference indices refIdxLXIvMC, refIdxLXIvMCShift and refIdxLXIvDC, the prediction list utilization flags predFlagLXIvMC, predFlagLXivMCShift and predFlagLXIvDC, and the motion vectors mvLXIvMC, mvLXIvMCShift and mvLXIvDC (with X being 0 or 1, respectively).
  5. Depending on view\_synthesis\_pred\_flag[ nuh\_layer\_id ], the following applies:
     + If view\_synthesis\_pred\_flag[ nuh\_layer\_id ] is equal to 0, the flag availableFlagVSP is set equal to 0.
     + Otherwise (view\_synthesis\_pred\_flag[ nuh\_layer\_id ] is equal to 1), the derivation process for a view synthesis prediction merge candidate as specified in subclause H.8.5.3.2.13 is invoked with the luma locations ( xCb, yCb ) as input and the outputs are the availability flag availableFlagVSP, the reference indices refIdxL0VSP and refIdxL1VSP, the prediction list utilization flags predFlagL0VSP and predFlagL1VSP, and the motion vectors mvL0VSP and mvL1VSP.
  6. Depending on mpi\_flag[ nuh\_layer\_id ], the following applies:
     + If mpi\_flag[ nuh\_layer\_id ] is equal to 0, the variable availableFlagT is set equal to 0.
     + Otherwise (mpi\_flag[ nuh\_layer\_id ] is equal to 1), the derivation process for the texture merging candidate as specified in subclause H.8.5.3.2.14 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH as the inputs and the outputs are the flag availableFlagT, the prediction utilization flags predFlagL0T and predFlagL1T, the reference indices refIdxL0T and refIdxL1T, and the motion vectors mvL0T and mvL1T.
  7. The merging candidate list, extMergeCandList, is constructed as follows:

i = 0  
 if( availableFlagT )  
 extMergeCandList[ i++ ] = T  
 if( availableFlagIvMC && ( !availableFlagT  | |  differentMotion( T, IvMC ) ) )  
 extMergeCandList[ i++ ] = IvMC  
 N = DepthFlag ? T : IvMC  
 if( availableFlagA1 && ( !availableFlagN  | |  differentMotion( N, A1 ) ) )  
 extMergeCandList[ i++ ] = A1 if( availableFlagB1 && ( !availableFlagN  | |  differentMotion( N, B1 ) ) )  
 extMergeCandList[ i++ ] = B1 if( availableFlagB0 )  
 extMergeCandList[ i++ ] = B0 (H‑91) if( availableFlagIvDC && ( !availableFlagA1  | |  differentMotion( A1, IvDC ) ) &&   
 ( !availableFlagB1  | |  differentMotion( B1, IvDC ) ) )  
 extMergeCandList[ i++ ] = IvDC  
 if( availableFlagVSP && !(mergeCandIsVspFlagA1 || mergeCandIsVspFlagB1 || mergeCandIsVspFlagB0 ) && !ic\_flag && iv\_res\_pred\_weight\_idx = = 0 )  
 extMergeCandList[ i++ ] = VSP  
 if( availableFlagA0 )  
 extMergeCandList[ i++ ] = A0 if( availableFlagB2 )  
 extMergeCandList[ i++ ] = B2 if( availableFlagIvMCShift && i < ( 5 + NumExtraMergeCand ) &&  
 ( !availableFlagIvMC  | |  differentMotion( IvMC, IvMCShift ) ) )  
 extMergeCandList[ i++ ] = IvMCShift

* 1. The variable availableFlagIvDCShift is set equal to 0, and when availableFlagIvMCShift is equal to 0, and i is less than ( 5 + NumExtraMergeCand ), the derivation process for the shifted disparity merging candidate as specified in subclause H.8.5.3.2.15 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, and the availability flags availableFlagN, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N, of every candidate N being in extMergeCandList, extMergeCandList, and i as the inputs and the outputs are the flag availableFlagIvDCShift, the prediction utilization flags predFlagL0IvDCShift and predFlagL1IvDCShift, the reference indices refIdxL0IvDCShift and refIdxL1IvDCShift, and the motion vectors mvL0IvDCShift and mvL1IvDCShift.
  2. The merging candidate list, extMergeCandList, is constructed as follows:

if( availableFlagIvDCShift )  
 extMergeCandList[ i++ ] = IvDCShift  
 j = 0  
 while( i < MaxNumMergeCand ) {(H‑92)  
 N = baseMergeCandList[ j++ ]  
 if( N != A1 && N != B1 && N != B0 && N != A0 && N != B2 )  
 extMergeCandList[ i++ ] = N  
 }

* 1. The variable N is set equal to extMergeCandList[ merge\_idx[ xOrigP ][ yOrigP ] ].
  2. The variable subPbMotionFlag is set equal to ( N = = IvMC ).
  3. The following assignments are made with X being replaced by 0 or 1:
     + 1. mvLX = subPbMotionFlag ? 0 : mvLXN (H‑93)
       2. refIdxLX = subPbMotionFlag ? −1 : refIdxLXN (H‑94)
       3. predFlagLX = subPbMotionFlag ? 0 : predFlagLXN (H‑95)
  4. When predFlagL0 is equal to 1 and predFlagL1 is equal to 1, and ( nOrigPbW + nOrigPbH ) is equal to 12, the following applies
     + 1. refIdxL1 = −1 (H‑96)
       2. predFlagL1 = 0 (H‑97)
  5. ~~The derivation process for a view synthesis prediction flag as specified in subclause H.8.5.3.2.17 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, the merge candidate indicator N as the inputs, and the output is the mergeCandIsVspFlag.~~
  6. The variable vspModeFlag is derived as specified in the following:
     1. vspModeFlag = mergeCandIsVspFlagN && !ic\_flag && ( iv\_res\_pred\_weight\_idx = = 0 ) (H‑98)
  7. The disparity availability flag ivpMvFlagLX is derived as follows (with X being replace by 0 or 1).
     + If DepthFlag is equal to 0 and one of the following conditions is true, ivpMvFlagLX is set equal to 1
       - predFlagLXIvMC && extMergeCandList[ merge\_idx[ xPb][ yPb ] ]  =  =  IvMC
       - predFlagLXIvMCShift && extMergeCandList[ merge\_idx[ xPb][ yPb ] ]  = =  IvMCShift
     + Otherwise, ivpMvFlagLX is set equal to 0.

H.8.5.3.2.17 Derivation process for a view synthesis prediction flag

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 merge candidate indicator N, specifying the merge candidate.

Outputs of this process are:

* a variable mergeCandIsVspFlagN specifying, whether the merge candidate is a view synthesis prediction merge candidate.

1. The variable mergeCandIsVspFlagN is derived as specified in the following:

* If N is equal to VSP, mergeCandIsVspFlagN is set equal to 1,
* Otherwise, if N is equal to A1, mergeCandIsVspFlagA1 is set equal to VspModeFlag[ xPb − 1 ][ yPb + nPbH − 1 ],
* Otherwise, if N is equal to B0 and ( yN >> Log2CtbSizeY ) is equal to ( yCb >> Log2CtbSizeY ), mergeCandIsVspFlagN is set equal to VspModeFlag[ xPb − 1 ][ yPb + nPbH ],
* Otherwise, if N is equal to B1 and ( yN >> Log2CtbSizeY ) is equal to ( yCb >> Log2CtbSizeY ), mergeCandIsVspFlagN is set equal to VspModeFlag[ xPb − 1 ][ yPb + nPbH ],
* Otherwise, mergeCandIsVspFlagN is set equal to 0.