Derivation process for luma motion vectors for merge mode

This process is only invoked when MergeFlag[ 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 flag ivpMvFlag, specifying, whether the current PU is coded using inter-view motion prediction,
* 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 flag dispDerivedDepthFlag, specifying, whether the current PU uses disparity derived depth,
* the variable dispDerivedDepthVal (when dispDerivedDepthFlag is equal to 1).

[Ed. (GT): In particular two things need to be check in this process: 1.) Are the limits on candidates in the list correct ( e.g. MaxNumMergeCand vs.  5 + NumExtraMergeCand  ) 2.) Is ( xOrigP, yOrigP ) and ( xPb, yPb ) used correctly in all places? ]

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 I.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.
     + Otherwise (N is not an element in baseMergeCandList), availableFlagN is set equal to 0.
  3. Depending on iv\_mv\_pred\_flag[ nuh\_layer\_id ] and DispAvailabilityIdc[ xPb ][ yPb ], the following applies:
     + If iv\_mv\_pred\_flag[ nuh\_layer\_id ] is equal to 0 or DispAvailabilityIdc[ xPb ][ yPb ] is not equal to DISP\_NONE, 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 I.8.5.3.2.10 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH as 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).
  4. Depending on view\_synthesis\_pred\_flag[ nuh\_layer\_id ], DispAvailabilityIdc[ xPb ][ yPb ], and , dbbp\_flag[ xPb ][ yPb ] the following applies:
     + If view\_synthesis\_pred\_flag[ nuh\_layer\_id ] is equal to 0, DispAvailabilityIdc[ xPb ][ yPb ] is equal to DISP\_NONE, or dbbp\_flag[ xPb ][ yPb ] is equal to 1, the flag availableFlagVSP is set equal to 0.
     + Otherwise (view\_synthesis\_pred\_flag[ nuh\_layer\_id ] is equal to 1, DispAvailabilityIdc[ xPb ][ yPb ] is not equal to DISP\_NONE, and dbbp\_flag[ xPb ][ yPb ] is equal to 0), the following applies:
       - A flag skipVSP is set equal to 0. And if one of the following is true, skipVSP is set equal to 1:
         * availableA1 && VspModeFlag[xPb – 1][yPb + nPbH – 1]
         * availableB1 && VspModeFlag[xPb + nPbW – 1][yPb – 1] && ( ( ( yPb – 1 ) >> Log2CtbSizeY ) == ( yCb >> Log2CtbSizeY) ) )
         * availableB0 && VspModeFlag[xPb + nPbW][yPb – 1] && ( ( ( yPb – 1 ) >> Log2CtbSizeY ) == ( yCb >> Log2CtbSizeY) ) )
         * availableA0 && VspModeFlag[xPb – 1][yPb + nPbH]
         * availableB2 && VspModeFlag[xPb – 1][yPb – 1] && ( ( ( yPb – 1 ) >> Log2CtbSizeY ) == ( yCb >> Log2CtbSizeY) ) )
       - If skipVSP is not equal to 1, the following applies:
         * the derivation process for a view synthesis prediction merge candidate as specified in subclause I.8.5.3.2.13 is invoked with the luma locations ( xPb, yPb ) and the variables nPbW and nPbH as inputs, 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.
  5. Depending on mpi\_flag[ nuh\_layer\_id ], the following applies:
     + If mpi\_flag[ nuh\_layer\_id ] is equal to 0, the variables availableFlagT and availableFlagD are set equal to 0.
     + Otherwise (mpi\_flag[ nuh\_layer\_id ] is equal to 1), the following applies:
       - The derivation process for inter layer predicted sub prediction block motion vector candidates as specified in subclause I.8.5.3.2.16 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, the variable refViewIdx being equal to −1, and the variable mvDisp being equal to ( 0, 0 ) as inputs, and the outputs are the prediction utilization flag predFlagLXT, the motion vector mvLXT and the reference indices refIdxLXT (with X being 0 or 1, respectively).
       - The flag availableFlagT is set equal to ( predFlagL0T | | predFlagL1T ).
       - The derivation process for the disparity derived merging candidates as specified in subclause I.8.5.3.2.19 is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH as inputs, and the outputs are the flag availableFlagD, the prediction utilization flag predFlagLXD, the reference index refIdxLXD, the motion vector mvLXD (with X being 0 or 1, respectively), and the variable dispDerivedDepthVal.
  6. The merging candidate list, extMergeCandList, is constructed as follows:

i = 0  
 if( availableFlagT )  
 extMergeCandList[ i++ ] = T  
 if( availableFlagD )  
 extMergeCandList[ i++ ] = D  
 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 (I‑103) if( availableFlagIvDC && ( !availableFlagA1  | |  differentMotion( A1, IvDC ) ) &&   
 ( !availableFlagB1  | |  differentMotion( B1, IvDC ) ) && ( i < ( 5 + NumExtraMergeCand ) ) )  
 extMergeCandList[ i++ ] = IvDC  
 if( availableFlagVSP && !ic\_flag && iv\_res\_pred\_weight\_idx = = 0 &&   
 i < ( 5 + NumExtraMergeCand ) )  
 extMergeCandList[ i++ ] = VSP  
 if( availableFlagA0 && i < ( 5 + NumExtraMergeCand ) )  
 extMergeCandList[ i++ ] = A0 if( availableFlagB2 && i < ( 5 + NumExtraMergeCand ) )  
 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, DepthFlag is equal to 0, and i is less than ( 5 + NumExtraMergeCand ), the derivation process for the shifted disparity merging candidate as specified in subclause I.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 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 ) {(I‑104)  
 N = baseMergeCandList[ j++ ]  
 if( N != A1 && N != B1 && N != B0 && N != A0 && N != B2 )  
 extMergeCandList[ i++ ] = N  
 }

* 1. The variable N is derived as specified in the following:
     + If ( nOrigPbW + nOrigPbH ) is equal to 12, the following applies:
       - 1. N = baseMergeCandList[ MergeIdx[ xOrigP ][ yOrigP ] ](I‑105)
     + Otherwise, ( ( nOrigPbW + nOrigPbH ) is not equal to 12 ), the following applies:
       - 1. N = extMergeCandList[ MergeIdx[ xOrigP ][ yOrigP ] ](I‑106)
  2. The derivation process for a view synthesis prediction flag as specified in subclause I.8.5.3.2.17 is invoked with the luma location ( xCb, yCb ), the luma location ( xPb, yPb ), the variables nPbW and nPbH, the merge candidate indicator N as the inputs, and the output is the mergeCandIsVspFlag.
  3. The variable vspModeFlag is derived as specified in the following:
     + 1. vspModeFlag = mergeCandIsVspFlag && !ic\_flag &&   
           ( iv\_res\_pred\_weight\_idx = = 0 ) && availableFlagVSP (I‑107)
  4. The variable subPbMotionFlag is derived as specified in the following:
     + 1. subPbMotionFlag = ( ( ( N = = IvMC ) && ( PartMode = = PART\_2Nx2N ) )  
           | | vspModeFlag ) && !dbbp\_flag (I‑108)
       2. [Note (FJ): This needs to be aligned in the software, as VSP is now also implemented as a special case of subPbMotionPrediction and DBBP does not support subPbMotionPrediction.]
  5. The following assignments are made with X being replaced by 0 or 1:
     + 1. mvLX = subPbMotionFlag ? 0 : mvLXN (I‑109)
       2. refIdxLX = subPbMotionFlag ? −1 : refIdxLXN (I‑110)
       3. predFlagLX = subPbMotionFlag ? 0 : predFlagLXN (I‑111)
  6. 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 (I‑112)
       2. predFlagL1 = 0 (I‑113)
  7. The disparity availability flag ivpMvFlag is derived as follows:
     + 1. ivpMvFlag = !DepthFlag && ( ( N  =  =  IvMC ) | | ( N  = =  IvMCShift ) ) (I‑114)
  8. The variable dispDerivedDepthFlag is derived as follows:
     + 1. dispDerivedDepthFlag = ( N  = =  D ) (I‑115)