I.8.5.3 Decoding process for prediction units in inter prediction mode

I.8.5.3.1 General

Inputs to this process are:

* a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
* a luma location ( xBl, yBl ) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block,
* a variable nCbS specifying the size of the current luma coding block,
* a variable nPbW specifying the width of the current luma prediction block,
* a variable nPbH specifying the width of the current luma prediction block,
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are:

* an (nCbSL)x(nCbSL) array predSamplesL of luma prediction samples, where nCSL is derived as specified below,
* an (nCbSC)x(nCbSC) array predSamplesCb of chroma prediction samples for the component Cb, where nCSC is derived as specified below,
* an (nCbSC)x(nCbSC) array predSamplesCr of chroma prediction samples for the component Cr, where nCSC is derived as specified below.

The variable nCbSL is set equal to nCbS and the variable nCbSC is set equal to nCbS  >>  1.

The decoding process for prediction units in inter prediction mode consists of the following ordered steps:

* 1. The derivation process for motion vector components and reference indices as specified in subclause I.8.5.3.2 is invoked with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ), the luma coding block size block nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH, and the prediction unit index partIdx as inputs, and the luma motion vectors mvL0 and mvL1, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1, the prediction list utilization flags predFlagL0 and predFlagL1 and the flags subPbMotionFlag and spModeFlag as outputs.
  2. Depending on subPbMotionFlag, spModeFlag, and dbbp\_flag[ xCb ][ yCb ] the following applies:
     1. [Ed.(GT): Can the case occur that both above flags are equal to 1? When yes this should be forbidden.]  
        [Note (FJ): The software already forbids enabling both, subPbMotionFlag and dbbp\_flag. This restriction is added to I.8.5.3.2.1 ]
     + If both subPbMotionFlag and dbbp\_flag[ xCb ][ yCb ] and spModeFlag are equal to 0, the decoding process for inter sample prediction as specified in subclause I.8.5.3.3.1 is invoked with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ), the luma coding block size block nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH, the luma motion vectors mvL0 and mvL1, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1, and the prediction list utilization flags predFlagL0 and predFlagL1 as inputs, and the inter prediction samples (predSamples) that are an (nCbSL)x(nCbSL) array predSamplesL of prediction luma samples and two (nCbSC)x(nCbSC) arrays predSamplesCr and predSamplesCr of prediction chroma samples, one for each of the chroma components Cb and Cr, as outputs.
     + Otherwise, if subPbMotionFlag is equal to 1, the decoding process for sub prediction block wise inter sample prediction as specified in subclause I.8.5.3.3.8 is invoked with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ), the luma coding block size block nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH as inputs, and the inter prediction samples (predSamples) that are an (nCbSL)x(nCbSL) array predSamplesL of prediction luma samples and two (nCbSC)x(nCbSC) arrays predSamplesCr and predSamplesCr of prediction chroma samples, one for each of the chroma components Cb and Cr, as outputs.
     + Otherwise, if dbbp\_flag[ xCb ][ yCb ] is equal to 1, the decoding process for depth based block partition wise inter sample prediction as specified in subclause I.8.5.3.3.9 is invoked with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ) ~~set to ( 0, 0 )~~, the luma coding block size nCbS, the luma prediction block width nPbW set to nCbS, the luma prediction block height nPbH set to nCbS, the luma motion vectors mvL0 and mvL1, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1, the prediction list utilization flags predFlagL0 and predFlagL1, and the variable partIdx.
     + Otherwise, if spModeFlag is equal to 1, the decoding process for depth based block partition wise inter sample prediction as specified in subclause I.8.5.3.3.9 is invoked with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ), the luma coding block size nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH, the luma motion vectors mvL0 and mvL1, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1, the prediction list utilization flags predFlagL0 and predFlagL1, and the variable partIdx set to 0.  
       The decoding process for depth based block partition wise inter sample prediction as specified in subclause I.8.5.3.3.9 is invoked a second time with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ), the luma coding block size nCbS, the luma prediction block width nPbW, the luma prediction block height nPbH, the luma motion vectors mvL0 and mvL1, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1, the prediction list utilization flags predFlagL0 and predFlagL1, and the variable partIdx set to 1.
       1. [Ed. (GT): In the current version of this draft predSamplesL, predSamplesCb and predSamplesCr are not defined for dbbp\_flag[ xCb ][ yCb ] is equal to 1. Reason for this is that assignment of predSamples of the Pbs to the Cb is broken in HEVC version 1. As workaround predicted samples for dbbp\_flag[ xCb ][ yCb ] equal to 1 are stored in PredSamplesDbbpL, PredSamplesDbbpCr and PredSamplesDbbpCb and assigned to the Cb in I.8.5.1.]

For use in derivation processes of variables invoked later in the decoding process, the following assignments are made for x = xBl..xBl + nPbW − 1 and y = yBl..yBl + nPbH − 1:

* 1. MvL0[ xCb + x ][ yCb + y ] = subPbMotionFlag ? SubPbMvL0[ xCb + x ][ yCb + y ] : mvL0 (I‑83)
  2. MvL1[ xCb + x ][ yCb + y ] = subPbMotionFlag ? SubPbMvL1[ xCb + x ][ yCb + y ] : mvL1 (I‑84)
  3. RefIdxL0[ xCb + x ][ yCb + y ] = subPbMotionFlag ? SubPbRefIdxL0[ xCb + x ][ yCb + y ] : refIdxL0 (I‑85)
  4. RefIdxL1[ xCb + x ][ yCb + y ] = subPbMotionFlag ? SubPbRefIdxL1[ xCb + x ][ yCb + y ] : refIdxL1 (I‑86)
  5. PredFlagL0[ xCb + x ][ yCb + y ] = subPbMotionFlag ? SubPbPredFlagL0[ xCb + x ][ yCb + y ] : predFlagL0 (I‑87)
  6. PredFlagL1[ xCb + x ][ yCb + y ] = subPbMotionFlag ? SubPbPredFlagL1[ xCb + x ][ yCb + y ] : predFlagL1 (I‑88)

I.8.5.3.2 Derivation process for motion vector components and reference indices

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 ( xBl, yBl ) of the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block,
* 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 chroma motion vectors mvCL0 and mvCL1,
* the reference indices refIdxL0 and refIdxL1,
* the prediction list utilization flags predFlagL0 and predFlagL1.
* the flag subPbMotionFlag, specifying, whether the motion data of the current PU has sub prediction block size motion accuracy,
* the flag spModeFlag, specifying whether the current PU uses segment-wise prediction.

Let ( xPb, yPb ) specify the top-left sample location of the current luma prediction block relative to the top-left luma sample of the current picture where xPb = xCb + xBl and yPb = yCb + yBl.

Let the variable currPic and ListX be the current picture and RefPicListX, with X being 0 or 1, of the current picture, respectively.

The function LongTermRefPic( aPic, aPb, refIdx, LX ), with X being 0 or 1, is defined as follows:

* If the picture with index refIdx from reference picture list LX of the slice containing prediction block aPb in the picture aPic was marked as "used for long term reference" at the time when aPic was the current picture, LongTermRefPic( aPic, aPb, refIdx, LX ) is equal to 1.
* Otherwise, LongTermRefPic( aPic, aPb, refIdx, LX ) is equal to 0.

The variables vspModeFlag, ivpMvFlag, subPbMotionFlag, dispDerivedDepthFlag and dispDerivedDepthVal, spModeFlag, refIdxS1, predFlagS1 and mvS1 are set equal to 0.

For the derivation of the variables mvL0 and mvL1, refIdxL0 and refIdxL1, as well as predFlagL0 and predFlagL1, the following applies:

* If MergeFlag[ xPb ][ yPb ] is equal to 1, the derivation process for luma motion vectors for merge mode as specified in subclause I.8.5.3.2.1 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 the luma motion vectors mvL0, mvL1, the reference indices refIdxL0, refIdxL1, and the prediction list utilization flags predFlagL0 and predFlagL1,, the flag ivpMvFlag, the flag vspModeFlag, the flag subPbMotionFlag, the flag dispDerivedDepthFlag and the variable dispDerivedDepthVal, the flag spModeFlag and the variables refIdxS1, predFlagS1 and mvS1.
* Otherwise, for X being replaced by either 0 or 1 in the variables predFlagLX, mvLX, and refIdxLX, in PRED\_LX, and in the syntax elements PuRefIdxLX and MvdLX, the following applies:
  + 1. The variables refIdxLX and predFlagLX are derived as follows:
       - * If InterPredIdc[ xPb ][ yPb ] is equal to PRED\_LX or PRED\_BI,

refIdxLX = PuRefIdxLX[ xPb ][ yPb ] (I‑89)

predFlagLX = 1 (I‑90)

* + - * + Otherwise, the variables refIdxLX and predFlagLX are specified by:

refIdxLX = −1 (I‑91)

predFlagLX = 0 (I‑92)

* + 1. The variable mvdLX is derived as follows:

mvdLX[ 0 ] = MvdLX[ xPb ][ yPb ][ 0 ] (I‑93)

mvdLX[ 1 ] = MvdLX[ xPb ][ yPb ][ 1 ] (I‑94)

* + 1. When predFlagLX is equal to 1, the derivation process for luma motion vector prediction in subclause 8.5.3.2.5 is invoked with the luma coding block location ( xCb, yCb ), the coding block size nCbS, the luma prediction block location ( xPb, yPb ), the variables nPbW, nPbH, refIdxLX, and the partition index partIdx as inputs, and the output being mvpLX.
    2. When predFlagLX is equal to 1, the luma motion vector mvLX is derived as follows:

uLX[ 0 ] = ( mvpLX[ 0 ] + mvdLX[ 0 ] + 216 ) % 216 (I‑95)

mvLX[ 0 ] = ( uLX[ 0 ] >= 215 ) ? ( uLX[ 0 ] − 216 ) : uLX[ 0 ] (I‑96)

uLX[ 1 ] = ( mvpLX[ 1 ] + mvdLX[ 1 ] + 216 ) % 216 (I‑97)

mvLX[ 1 ] = ( uLX[ 1 ] >= 215 ) ? ( uLX[ 1 ] − 216 ) : uLX[ 1 ] (I‑98)

NOTE – The resulting values of mvLX[ 0 ] and mvLX[ 1 ] as specified above will always be in the range of −215 to 215 − 1, inclusive.

When ChromaArrayType is not equal to 0 and predFlagLX, with X being 0 or 1, is equal to 1, the derivation process for chroma motion vectors in subclause 8.5.3.2.9 is invoked with mvLX as input, and the output being mvCLX.

For use in derivation processes of variables invoked later in the decoding process, the following assignments are made for x = xPb.. ( xPb + nPbW − 1 ), y = yPb..( yPb + nPbH− 1 ) (with X being either 0 or 1):

* 1. IvpMvFlag[ x ][ y ] = ivpMvFlag (I‑99)  
     VspModeFlag[ x ][ y ] = vspModeFlag (I‑100)  
     DispDerivedDepthFlag[ x ][ y ]  = dispDerivedDepthFlag (I‑101)  
     DispDerivedDepthVal[ x ][ y ] = dispDerivedDepthVal (I‑102)

**I.8.5.3.2.1 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.

…

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).
* the flag spModeFlag, specifying whether the current PU uses segment-wise prediction,
* the variables refIdxS1, predFlagS1 and mvS1 (when spModeFlag is equal to 1).

…

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 and Col, 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 depth\_based\_blk\_part\_flag[ nuh\_layer\_id ], DispAvailabilityIdc[ xPb ][ yPb ], and , dbbp\_flag[ xPb ][ yPb ] the following applies:
     1. If depth\_based\_blk\_part\_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 availableFlagSP is set equal to 0.

Otherwise (depth\_based\_blk\_part\_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 derivation process for a segment-wise prediction merge candidate as specified in subclause I.8.5.3.2.20 is invoked with the luma locations ( xPb, yPb ), the variables nPbW and nPbH, the baseMergeCandList and the availability flags availableFlagN (with N being A1, B1, B0, A0, B2 and Col) as inputs, and the outputs are the availability flag availableFlagSP, the reference indices refIdxL0SP, refIdxL1SP and refIdxS1, the prediction list utilization flags predFlagL0SP, predFlagL1SP and predFlagS1, and the motion vectors mvL0SP, mvL1SP and mvS1.

* 1. 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 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.

* 1. 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.
  2. 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( availableFlagSP && !ic\_flag && iv\_res\_pred\_weight\_idx = = 0 &&   
 i < ( 5 + NumExtraMergeCand ) )  
 extMergeCandList[ i++ ] = SP  
 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

I.8.5.3.2.20 Derivation process for a segment-wise prediction merge candidate

Inputs to this process are:

* 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,
* two variables nPbW and nPbH specifying the width and the height of the current prediction block.
* the merge candidate list, mergeCandList,
* the flags availableFlagN, with N being A1, B1, B0, A0, B2 and Col, specifying whether the corresponding merging candidate is available.

Outputs of this process are

* the availability flag availableFlagSP whether the SP merge candidate is available,
* the reference indices refIdxL0SP, refIdxL1SP and refIdxS1,
* the prediction list utilization flags predFlagL0SP, predFlagL1SP and predFlagS1,
* the motion vectors mvL0SP, mvL1SP and mvS1.

The derivation process for a depth predicted contour pattern as specified in subclause I.8.5.8 is invoked with the sampling interval sampInt equal to 1, the sample location ( xTb, yTb ) equal to ( xPb, yPb ), and the block size ( nTb, nTbH ) equal to ( nPbW, nPbH ) as inputs, and the output is a binary partition pattern segMask[ x ][ y ].

[Note: This requires the partition pattern process to allow non-square block sizes.]

The array segMvSrc[ X ] with X = 0..1 is set equal to INVALID before the following process is invoked:

X= segMask[ 0 ][ nPbH-1 ]  
if( availableFlagA0 && segMvSrc[ X ] == INVALID )  
 segMvSrc[ X ] = A0  
X= segMask[ nPbW-1 ][ 0 ]  
if( availableFlagB0 && segMvSrc[ X ] == INVALID )  
 segMvSrc[ X ] = B0X= segMask[ 0 ][ 0 ]  
if( availableFlagB2 && segMvSrc[ X ] == INVALID )  
 segMvSrc[ X ] = B2  
X= segMask[ nPbW-1 ][ nPbH-1 ]  
if( availableFlagCol && segMvSrc[ X ] == INVALID  
 segMvSrc[ X ] = Col  
X= segMask[ 0 ][ nPbH-1 ]  
if( availableFlagA1 && segMvSrc[ X ] == INVALID )  
 segMvSrc[ X ] = A1  
X= segMask[ nPbW-1 ][ 0]  
if( availableFlagB1 && segMvSrc[ X ] == INVALID )  
 segMvSrc[ X ] = B1  
  
segmentMvAvailableFlag = segMvSrc[ 0 ] != INVALID && segMvSrc[ 1 ] != INVALID  
if( segmentMvAvailableFlag ) {  
 Y = segMvSrc[ 0 ]  
 Z = predFlagL0Y ? 0 : 1  
 refIdxLZSP = refIdxLZY  
 predFlagLZSP = 1  
 mvLZSP = mvLZY  
   
 Y = segMvSrc[ 1 ]   
 refIdxS1 = predFlagL0Y ? refIdxL0Y : refIdxL1Y  
 predFlagS1 = predFlagL0Y ? 0 : 1  
 mvS1 = predFlagL0Y ? mvL0Y : mvL1Y  
  
 if( Z != predFlagS1 || refIdxLZSP != refIdxS1 || mvLZSP != mvS1 )  
 availableFlagSP = 1  
}

I.8.5.3.3.9 Decoding process for depth based block partition wise inter sample prediction

Inputs to this process are:

* a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
* a luma location ( xBl, yBl ) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block
* 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
* the luma motion vectors mvL0 and mvL1,
* the chroma motion vectors mvCL0 and mvCL1,
* the reference indices refIdxL0 and refIdxL1,
* the prediction list utilization flags, predFlagL0, and predFlagL1,
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.

1. The variable nCbSL is set equal to nCbS and the variable nCbSC is set equal to nCbS  >>  1.

The decoding process for inter sample prediction as specified in subclause I.8.5.3.3.1 is invoked with the luma coding block location ( xCb, yCb ), the luma prediction block location ( xBl, yBl ) ~~set to ( 0, 0 )~~, the luma coding block size nCbS, the luma prediction block width nPbW ~~set to nCbS~~, the luma prediction block height nPbH ~~set to nCbS~~, the luma motion vectors mvL0 and mvL1, the chroma motion vectors mvCL0 and mvCL1, the reference indices refIdxL0 and refIdxL1, and the prediction list utilization flags predFlagL0 and predFlagL1 as inputs, and the inter prediction samples (predSamples) that are an (nPbW)x(nPbH) array predSamplesL of prediction luma samples and two (nPbW/2)x(nPbH/2) arrays predSamplesCr and predSamplesCr of prediction chroma samples, one for each of the chroma components Cb and Cr, as outputs.

The derivation process for a depth predicted contour pattern as specified in subclause I.8.5.8 is invoked with the sampling interval sampInt equal to 1, the sample location ( xTb, yTb ) equal to ( xCb+xBl, yCb+yBl ), and the block size nTbS equal to (nPbW)x(nPbH) as inputs, and the output is a binary partition pattern segMask[ x ][ y ].

[Note: The derivation of the PU’s segmentation mask need to allow non-square shapes.]

The arrays PredSamplesDbbpL, PredSamplesDbbpCb and PredSamplesDbbpCr are modified as follows:

for ( y = 0; y < nPbH; y++ )  
 for( x = 0; x < nPbW; x++ ) {  
 if( segMask[ x ][ y ] = = ( partIdx ! = segMask[ 0 ][ 0 ] ) )  
 PredSamplesDbbpL[ x ][ y ] = predSamplesL[ x ][ y ]  
 if( ( x % 2 = = 0 ) && ( y % 2 = = 0 ) ) {  
 PredSamplesDbbpCb[ x / 2 ][ y / 2 ] = predSamplesCb[ x / 2 ][ y / 2 ]  
 PredSamplesDbbpCr[ x / 2 ][ y / 2 ] = predSamplesCr[ x / 2 ][ y / 2 ]  
 }  
 }

When partIdx is equal to 1, the arrays PredSamplesDbbpL, PredSamplesDbbpCb and PredSamplesDbbpCr are modified as follows:

* The derivation process for contour boundary filtered samples as specified in subclause I.8.5.3.3.9.1 is invoked with, the luma coding block size block nCbSL, the current coding block size nCbSX set equal to (nPbW)x(nPbH), the array segMask, the array predSamples of prediction samples equal to PredSamplesDbbpL as inputs and the output is assigned to the array PredSamplesDbbpL of luma prediction samples.
* The derivation process for contour boundary filtered samples as specified in subclause I.8.5.3.3.9.1 is invoked with, the luma coding block size block nCbSL, the current coding block size nCbSX set equal to (nPbW/2)x(nPbH/2), the array segMask, the array predSamples of prediction samples equal to PredSamplesDbbpCb as inputs and the output is assigned to the array PredSamplesDbbpCb of luma prediction samples.

The derivation process for contour boundary filtered samples as specified in subclause I.8.5.3.3.9.1 is invoked with, the luma coding block size block nCbSL, the current coding block size nCbSX set equal to (nPbW)x(nPbH), the array segMask, the array predSamples of prediction samples equal to PredSamplesDbbpCr as inputs and the output is assigned to the array PredSamplesDbbpCr of luma prediction samples.