G.7.3.6.1 General slice segment header syntax

|  |  |
| --- | --- |
| … |  |
| if( slice\_temporal\_mvp\_enabled\_flag ) { |  |
| if( slice\_type = = B ) |  |
| **collocated\_from\_l0\_flag** | u(1) |
| if( ( collocated\_from\_l0\_flag && num\_ref\_idx\_l0\_active\_minus1 > 0 ) | |  ( !collocated\_from\_l0\_flag && num\_ref\_idx\_l1\_active\_minus1 > 0 ) ) |  |
| **collocated\_ref\_idx** | ue(v) |
| if( nuh\_layer\_id > 0 ){ |  |
| **aux\_collocated\_flag** | u(1) |
| if(aux\_collocated\_flag) |  |
| for( i=0; i<2; i++){ |  |
| if( slice\_type = = B ) |  |
| **aux\_collocated\_from\_l0\_flag**[i] | u(1) |
| if( (aux\_collocated\_from\_l0\_flag[i] && num\_ref\_idx\_l0\_active\_minus1 > 0 ) | | ( ! aux\_collocated\_from\_l0\_flag[i] && num\_ref\_idx\_l1\_active\_minus1 > 0 ) ) |  |
| **aux\_collocated\_ref\_idx**[i] | ue(v) |
| } |  |
| } |  |
| } |  |
| … |  |

G.7.4.7.1 General slice segment header semantics

**aux\_collocated\_flag** equal to 1 specifies that the collocated picture should be modified according to the auxiliary collocated pictures. aux\_collocated\_flag equal to 0 specifies that the collocated picture should not be modified. When aux\_collocated\_flag is not present, it is inferred to be equal to 0.

**aux\_collocated\_from\_l0\_flag**[i] equal to 1 specifies that the ith auxiliary collocated picture is derived from reference picture list 0. aux\_collocated\_from\_l0\_flag[i] equal to 0 specifies that the ith auxiliary collocated picture is derived from reference picture list 1. When aux\_collocated\_from\_l0\_flag[i] is not present, it is inferred to be equal to 1. When aux\_collocated\_from\_l0\_flag[i] is not present, it is inferred to be equal to 1.

**aux\_collocated\_ref\_idx**[i] specifies the reference index of the ith auxiliary collocated picture.

When slice\_type is equal to P or when slice\_type is equal to B and aux\_collocated\_from\_l0\_flag[i] is equal to 1, aux\_collocated\_ref\_idx[i] refers to a picture in list 0, and the value of aux\_collocated\_ref\_idx[i] shall be in the range of 0 to num\_ref\_idx\_l0\_active\_minus1, inclusive.

When slice\_type is equal to B and aux\_collocated\_from\_l0\_flag[i] is equal to 0, aux\_collocated\_ref\_idx[i] refers to a picture in list 1, and the value of aux\_collocated\_ref\_idx[i] shall be in the range of 0 to num\_ref\_idx\_l1\_active\_minus1, inclusive.

It is a requirement of bitstream conformance that the picture referred to by aux\_collocated\_from\_l0\_flag[i] shall be the same for all slices of a coded picture.

The variable auxColPic[i] is set equal to RefPicList0[ aux\_collocated\_ref\_idx[i] ] if aux\_collocated\_from\_l0\_flag[i] is equal to 1; Otherwise, the variable auxColPic[i] is set equal to RefPicList1[ aux\_collocated\_ref\_idx[i] ]

F8.3 Slice decoding processes

F.8.3.5 Decoding process for collocated picture modification

This process is invoked at the beginning of the decoding process when aux\_collocated\_flag is equal to 1.

A variable colPic is derived as

* If slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 0, colPic is set equal to RefPicList1[ collocated\_ref\_idx ].
* Otherwise (slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 1 or slice\_type is equal to P), colPic is set equal to RefPicList0[ collocated\_ref\_idx ].

colPicOrignal is a picture whose information is copied from colPic. colPic should be set equal to colPicOrignal when the slice finishes decoding.

For each 4x4 block in colPic with ( xCb, yCb ) specifying the top-left sample of the block relative to the top‑left luma sample of colPic, the modification process for a block in the collocated picture as specified in subclause F.8.3.5.1 is invoked with location ( xCb, yCb ), colPic, colPicOrignal, and a block size number 4 as the inputs.

F.8.3.5.1 Modification process for a block in the collocated picture

Inputs to this process are:

* a luma location ( xCb, yCb ) specifying the top-left sample a block relative to the top-left luma sample of colPic,
* a variable colPic specifying the collocated picture,
* a variable colPicOrignal specifying the original collocated picture,
* a variable nCS specifying the block size.

The variable curBlock is set equal to a nCS by nCS block with ( xCb, yCb ) as its top-left sample of colPic.

For X from 0 to 1 inclusively, PredFlagLX[ xCb][ yCb ], and RefIdxLX[ xCb][ yCb ] of colPic are set equal to 0 and -1 respectively.

curBlock is set as coded in an intra prediction mode in colPic.

The following applies sequentially:

1. For i from 0 to 1 inclusively, the derivation process for a disparity vector as specified in subclause F.8.3.5.2 is invoked with the location ( xCb, yCb ), colPic, a size number equal to nCS, ViewIdx (colPic) and i as inputs, and the outputs are the disparity vector mvDisp[i], the flag dvAvailFlag[i], the reference index auxRefIdxLX[i], the motion vector auxMvLX[i] with X being 0 to 1.
2. For i from 0 to 1 inclusively, if dvAvailFlag[i] is equal to 1, the following applies,
   1. The derivation process for a temporal inter-view motion vector as specified in subclause F.8.3.5.3 is invoked with the location ( xCb, yCb ), a width number equal to nCS, a height number equal to nCS, the disparity vector mvDisp[i], and the original collocated picture colPicOriginal as inputs, and the outputs are a flag availableFlagLXInterView, a temporal inter-view motion vector mvLXInterView and a reference index refIdxLX with X being 0 and 1.
   2. If availableFlagL0InterView or availableFlagL1InterView is equal to 1, the following applies,
      1. For X from 0 to 1 inclusive, PredFlagLX[ xCb][ yCb ], MvLX[ xCb][ yCb ], and RefIdxLX[ xCb][ yCb ] of colPic are set equal to availableFlagLXInterView, mvLXInterView, and refIdxLX respectively.
      2. curBlock is set as coded in an inter prediction mode.
      3. The whole decoding process of this sub-clause terminates.
3. For X from 0 to 1 inclusively, the following applies,
   1. A variable bMvFound is set equal to 0
   2. For i from 0 to 1 inclusively, the following applies when bMvFound is equal to 0,
      1. If auxRefIdxLX[i] is not equal to -1, the following applies:
         1. the derivation process for a scaled motion vector as specified in subclause F.8.3.5.4 is invoked with the reference list X, the reference index auxRefIdxLX[i], the motion vector auxMvLX[i], curBlock, auxColPic[i], and colPic as its inputs, and the output is a motion vector auxMvScaled.
         2. PredFlagLX[ xCb][ yCb ], MvLX[ xCb][ yCb ], and RefIdxLX[ xCb][ yCb ] of colPic are set equal to 1, auxMvLX[i] and auxRefIdxLX[i] respectively.
         3. curBlock is set as coded in an inter prediction mode.
         4. bMvFound is set equal to 1.

F.8.3.5.2 Derivation process for a disparity vector

Inputs to this process are:

* a luma location ( xCb, yCb ) specifying the top-left sample a block relative to the top-left luma sample of colPic,
* a variable colPic specifying the collocated picture.
* a variable nCbS specifying the size of the current luma coding block,
* a variable targetViewId,
* a variable auxIdx specifying the index of the auxiliary collocated picture.

Outputs of this process are:

* the disparity vector mvDisp,
* the availability flag availableFlag,
* the reference index auxRefIdxLX, with X being 0 to 1,
* the motion vector auxMvLX with X being 0 to 1.

The luma location ( xCCtr , yCCtr) specifying the centre position of the current luma coding block is derived as follows:

* 1. xCCtr = xCb + ( nCbS  >>  1 )
  2. yCCtr = yCb + ( nCbS  >>  1 )

The flag availableFlag is set equal to 0, and mvDisp is set equal to ( 0, 0 ).

The variable colPu is the prediction unit in auxColPic[auxIdx] covering the position ( ( xCCtr  >>  4 )  <<  4 ,  ( yCCtr  >>  4 )  <<  4 ).

The position ( xPCol, yPCol ) is set equal to the position of the top-left sample of colPu relative to the top-left luma sample of the auxColPic[auxIdx].

For each X from 0 to 1, the variables auxRefIdxLX, and auxMvLX are set equal to the variables RefIdxLX[ xPCol ][ yPCol ], and MvLX[ xPCol ][ yPCol ] of auxColPic[auxIdx], respectively. RefIdxLX[ xPCol ][ yPCol ] should be set equal to -1 if PredFlagLX[ xPCol ][ yPCol ] is 0 in auxColPic[auxIdx].

The following ordered steps apply and the whole decoding process of this sub-clause terminates once availableFlag is set to 1.

* 1. For each X from 0 to 1, inclusive, the following applies:
     + When colPu is not coded in an intra prediction mode and PredFlagLX[ xPCol ][ yPCol ] of auxColPic[auxIdx] is equal to 1, the following applies:
       - refPicAux is set to be the picture with reference index RefIdxLX [ xPCol ][ yPCol ] in the reference picture list RefPicListX of the slice containing prediction block colPu in the picture auxColPic[auxIdx].
       - When targetViewId is equal to ViewIdx(refPicAux), the following applies:

mvDisp = candMV[ xPCol ][ yPCol ]

availableFlag = 1

F.8.3.5.3 Derivation process for a temporal inter-view motion vector

Inputs to this process are:

* a luma location ( xPb, yPb ) of the top-left sample of a luma prediction block relative to the top-left luma sample of the collocated picture,
* two variables nPbW and nPbH specifying the width and the height of the current luma prediction block,
* a disparity vector mvDisp.
* a picture ivRefPic

Outputs of this process are:

* a flag availableFlagLXInterView specifying whether the temporal inter-view motion vector is available with X being 0 or 1,
* a temporal inter-view motion vector mvLXInterView with X being 0 or 1,
* a reference index refIdxLX specifying a reference picture in the reference picture list RefPicListLX with X being 0 or 1.

For X from 0 to 1, inclusively,the flag availableFlagLXInterView is set equal to 0, the variable refIdxLX is set equal to –1, and both components of mvLXInterView are set equal to 0.

The reference layer luma location ( xRef, yRef ) is derived by

* 1. xRefFull = xPb + ( nPbW >> 1 ) + ( ( mvDisp[ 0 ] + 2 ) >> 2 )
  2. yRefFull = yPb + ( nPbH >> 1 ) + ( ( mvDisp[ 1 ] + 2 ) >> 2 )
  3. xRef = Clip3( 0, PicWidthInSamplesL − 1, ( xRefFull >> 3 ) << 3 )
  4. yRef = Clip3( 0, PicHeightInSamplesL − 1, ( yRefFull >> 3 ) << 3 )

The variable ivRefPb specifies the luma prediction block covering the location given by ( xRef, yRef ) inside the inter-view reference picture specified by ivRefPic.

The luma location ( xIvRefPb, yIvRefPb ) is set equal to the top-left sample of the inter-view reference luma prediction block specified by ivRefPb relative to the top-left luma sample of the reference picture specified by ivRefPic.

When ivRefPb is not coded in an intra prediction mode, for X from 0 to 1 inclusive, availableFlagLXInterView, mvLXInterView, and refIdxLX are set equal to PredFlagLX[ xIvRefPb ][ yIvRefPb ], MvLX[ xIvRefPb ][ yIvRefPb ], and RefIdxLX[ xIvRefPb ][ yIvRefPb ] of ivRefPic, respectively.

F.8.3.5.4 Derivation process for a scaled motion vector

Inputs to this process are:

* a reference list auxRefPicList,
* a reference index auxRefIdx,
* a motion vector auxMv,
* a block CurrPb,
* a picture auxPic,
* a picture colPic.

Outputs of this process are:

* the motion vector auxMvScaled.

The variable auxRefPic is set to be the picture with reference index auxRefIdx in the reference picture list auxRefPicList of the slice containing block currPb in the picture auxPic. The variable colRefPic is set to be the picture with reference index 0 in the reference picture list auxRefPicList of the slice containing block currPb in the picture colPic. If colRefPic does not exist, it should be set to be the picture with reference index 0 in the reference picture list 0 of the slice containing block currPb in the picture colPic. The following applies:

auxPocDiff = DiffPicOrderCnt( auxPic, auxRefPic )

colPocDiff = DiffPicOrderCnt( colPic, colRefPic )

* + - * If colPocDiff is equal to currPocDiff or auxPocDiff is equal to 0, mvLXCol is derived as follows:

auxMvScaled = auxMv.

* + - * Otherwise, mvLXCol is derived as a scaled version of the motion vector mvCol as follows:

tx = ( 16384 + ( Abs( td )  >>  1 ) ) / td

distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 )  >>  6 )

auxMvScaled =  Clip3( −32768, 32767, Sign( distScaleFactor \* auxMv) \*   
 ( ( Abs( distScaleFactor \* auxMv) + 127 )  >>  8 ) )

where td and tb are derived as follows:

td = Clip3( −128, 127, auxPocDiff )

tb = Clip3( −128, 127, colPocDiff )