J.8.3.1.1 Depth-based derivation process for luma motion vectors for skipped macroblocks in P and SP slices

This process is invoked when mb\_type is equal to P\_Skip, nal\_unit\_type is equal to 21, DepthFlag is equal to 0, dmvp\_flag is equal to 1 and MbVSSkipFlag is equal to 0.

Outputs of this process are:

– the motion vector mvL0,

– the reference index refIdxL0.

For the derivation of the motion vector mvL0 and refIdxL0 of a P\_Skip macroblock type, the following ordered steps are specified:

1. The process specified in subclause J.8.3.1.5 is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType set equal to "na", and listSuffixFlag equal to 0 as input and the output is assigned to the motion vectormvL0 and the reference index refIdxL0 .
2. When refIdxL0 is equal to -1, the following applies:

- The reference index refIdxL0 for a skipped macroblock is derived as:

refIdxL0 = 0.

- the derivation process for luma motion vector prediction in subclause J.8.3.1.7 is invoked with mbPartIdx = 0, subMbPartIdx = 0, refIdxL0, and currSubMbType = “na” as the inputs and the output being mvL0.

1. J.8.3.1.3 Derivation process for luma motion vectors for B\_Skip, B\_Direct\_16x16, and B\_Direct\_8x8

Inputs to this process are current macroblock partition index mbPartIdx and subMbPartIdx.

Outputs of this process are the reference indices refIdxL0, refIdxL1, the motion vectors mvL0 and mvL1, the motion vector count variable subMvCnt, and the prediction list utilization flags, predFlagL0 and predFlagL1.

For the derivation of output, the following ordered steps are specified:

1. Let the variable currSubMbType be set equal to sub\_mb\_type[ mbPartIdx ].
2. The process specified in subclause J.8.3.1.5 is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType and listSuffixFlag set equal to 0 as input and the output is assigned to the motion vector mvL0 and the reference index refIdxL0.
3. The process specified in subclause J.8.3.1.5 is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType and listSuffixFlag set equal to 1 as input and the output is assigned to the motion vectormvL1 and the reference indices refIdxL1.

4. When both reference indices refIdxL0 and refIdxL1 are equal to -1, the following applies:

- The reference indices refIdxL0 and refIdxL1 are derived by:

refIdxL0 = MinPositive(refIdxL0A, minPositive(refIdxL0B, refIdxI0C)).

refIdxL1 = MinPositive(refIdxL1A, minPositive(refIdxL1B, refIdxI1C)).

where

MinPositive(x, y) =

- When both reference indices refIdxL0 and refIdxL1 are less than 0,

refIdxL0 = 0

refIdxL1 = 0

- the derivation process for luma motion vector prediction in subclause J.8.3.1.7 is invoked with mbPartIdx = 0, subMbPartIdx = 0, refIdxLX (with X being 0 or 1), and currSubMbType as the inputs and the output being mvLX.

J.8.3.1.4 Derivation process for the disparity vector and the inter-view reference

Inputs to this process are depth reference view component depthPic, the location of a top-left sample ( dbx1, dby1 ) of a partition and the listSuffixFlag.

Outputs of this process are a picture InterViewPic, an offset vector dv and a variable InterViewAvailable

Set InterViewAvailableequal to 0.

The following applies to derive an inter-view reference picture or inter-view only reference picture, InterViewPic, with X set to 1 when listFuffixFlag is 1 or 0 otherwise:

for( cIdx = 0;cIdx<num\_ref\_idx\_l0\_active\_minus1 + 1 && !InterViewAvailable; cIdx ++)  
 if ( view order index of RefPicList0[ cIdx ] is equal to 0) {  
 InterViewPic = RefPicList0[ cIdx ]  
 InterViewAvailable = 1  
 }

When InterViewAvailable is equal to 1, the following steps apply in order.

– The process specified in subclause  is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType set equal to "na", and listSuffixFlag set equal to 0 as input and with reference indices refIdxCandL0[ i ] and the motion vectors mvCandL0[ i ] as outputs with i equal to 0, 1, and 2 corresponding to neighbouring partition A, B, and C, respectively.

– The process specified in subclause  is invoked with mbPartIdx set equal to 0, subMbPartIdx set equal to 0, currSubMbType set equal to "na", and listSuffixFlag set equal to 1 as input and with reference indices refIdxCandL1[ i ] and the motion vectors mvCandL1[ i ] as outputs with i equal to 0, 1, and 2 corresponding to neighbouring partition A, B, and C, respectively.

The variable dv is derived as specified by the following ordered steps:

- Set the DvAvailable[i] with i equal to 0, 1, and 2 corresponding to neighbouring partition A, B, and C, respectively, as follows

for( i = 0;i<3; i ++) {  
 if (view order index of RefPicList0[ refIdxCandLX[ i ] ] is equal to 0 ) {  
  
 DvAvailable[i] = 1  
 }

- When one and only one of DvAvailable[0], DvAvailable[1] , DvAvailable[2] is equal to 1,

dv[0] = mvCandLX[i ][0]

dv[1] = mvCandLX[i ][1]

- Otherwise, the following steps applies in order.

1. The variable maxDepth is specified as follows:

maxDepth = INT\_MIN  
for( j = 0; j < partHeight; j+=(partHeight-1) )  
 for( i = 0; i < partWidth; i+=(partWidth-1))  
 if( depthPic[ dbx1 + i, dby1 + j ] > maxDepth ) maxDepth = depthPic[ dbx1 + i, dby1 + j ]

2. The variable disp is specified as follows:  
index = ViewIdTo3DVAcquisitionParamIndex( view\_id of the current view )  
refIndex = ViewIdTo3DVAcquisitionParamIndex( view\_id of the InterViewPic )

disp[ 0 ] = Disparity( NdrInverse[maxDepth], ZNear[ dps\_id, index ], ZFar[dps\_id, index ],  
 FocalLengthX[dps\_id, index ], AbsTX[ index ] – AbsTX[ refIndex ] )  
disp[ 1 ] = 0

3. When DvAvailable[i] is equal to 0,

mvCandLX[i] = disp

4. Each component of the variable dv is given by the median of the corresponding vector components of the motion vector mvCandLX[0], mvCandLX[1], and mvCandLX[2]: dv[0] = Median(mvCandLX[0][0], mvCandLX[1][0], and mvCandLX[2][0])

dv[1] = Median(mvCandLX[0][1], mvCandLX[1][1], and mvCandLX[2][1])