J.8.3.1.7 Derivation process for luma motion vector prediction

Inputs to this process are:

– the macroblock partition index mbPartIdx,

– the sub-macroblock partition index subMbPartIdx,

– the reference index of the current partition refIdxLX (with X being 0 or 1),

– the variable currSubMbType.

Output of this process is the prediction mvpLX of the motion vector mvLX (with X being 0 or 1).

The specifications of subclause 8.4.1.3 apply with the following changes.

– The following additional sentence is applied

If refIdxLX is not equal to refIdxLXN for any N = A, B, or C and X equal to 0 or 1, the following applies:

mbAddrN\mbPartIdxN\subMbPartIdxN is marked as not available  
refIdxLXN = -1  
mvLXN[ 0 ] = 0  
mvLXN[ 1 ] = 0

after the following paragraph in subclause 8.4.1.3:

The derivation process for the neighbouring blocks for motion data in subclause  is invoked with mbPartIdx, subMbPartIdx, currSubMbType, and listSuffixFlag = X (with X being 0 or 1 for refIdxLX being refIdxL0 or refIdxL1, respectively) as the input and with mbAddrN\mbPartIdxN\subMbPartIdxN, reference indices refIdxLXN and the motion vectors mvLXN with N being replaced by A, B, or C as the output.

– The following additional sentence is applied

– Otherwise, If refIdxLX is a reference index to an inter-view reference component or an inter-view only reference component, the depth-based derivation process for median luma inter-view motion vector prediction in subclause is invoked with mbAddrN\mbPartIdxN\subMbPartIdxN, mvLXN, refIdxLXN with N being replaced by A, B, or C, and refIdxLX as the inputs and the output is assigned to the motion vector predictor mvpLX.

– Otherwise, If refIdxLX is a reference index to an inter reference component or an inter only reference component, the depth-based derivation process for median luma temporal motion vector prediction in subclause J.8.3.1.9 is invoked with mbAddrN\mbPartIdxN\subMbPartIdxN, mvLXN, refIdxLXN with N being replaced by A, B, or C, and refIdxLX as the inputs and the output is assigned to the motion vector predictor mvpLX.

after the following paragraph in subclause 8.4.1.3

– Otherwise, if MbPartWidth( mb\_type ) is equal to 8, MbPartHeight( mb\_type ) is equal to 16, mbPartIdx is equal mvpLX = mvLXCto 1, and refIdxLXC is equal to refIdxLX, the motion vector predictor mvpLX is derived by:

mvpLX = mvLXC

J.8.3.1.8 Depth-based derivation process for median luma inter-view motion vector prediction

Inputs to this process are:

– the neighbouring partitions mbAddrN\mbPartIdxN\subMbPartIdxN (with N being replaced by A, B, or C),

– the motion vectors mvLXN (with N being replaced by A, B, or C) of the neighbouring partitions,

– the reference indices refIdxLXN (with N being replaced by A, B, or C) of the neighbouring partitions,,

– the reference index refIdxLX of the current partition.

Output of this process is the motion vector prediction mvpLX.

When either partition mbAddrN\mbPartIdxN\subMbPartIdxN is not available or refIdxLXN is not equal to refIdxLX, mvLXN is derived as specified by the following ordered steps:

1. The inverse macroblock scanning process as specified in subclause 6.4.1is invoked with CurrMbAddr as the input and the output is assigned to ( x1, y1 ).

2. The inverse macroblock partition scanning process specified in subclause 6.4.2.1 is invoked with mbPartIdx as the input and the output assigned to ( dx1, dy1 ).

3. The inverse sub-macroblock partition scanning process specified in subclause 6.4.2.2 is invoked with mbPartIdx and subMbPartIdx as the input and the output assigned to ( dx2, dy2 ).

4. The modification process of inter-view motion vector in median luma motion vector prediction as specified in subclause .1 is invoked with depthPic being equal to DepthRefPicList0[ refIdxL0 ], dbx1 being equal to x1 + dx1 + dx2, dby1 being equal to y1 + dy1 + dy2, and mv being equal to mvL0 as inputs and the output is assigned to the motion vector mvLXN.

Each component of the motion vector prediction mvpLX is given by the median of the corresponding vector components of the motion vector mvLXA, mvLXB, and mvLXC:

mvpLX[ 0 ] = Median( mvLXA[ 0 ], mvLXB[ 0 ], mvLXC[ 0 ] ) (J8-XX)

mvpLX[ 1 ] = Median( mvLXA[ 1 ], mvLXB[ 1 ], mvLXC[ 1 ] ) (J8-XX)

J.8.3.1.9 Depth-based derivation process for median luma temporal motion vector prediction

Inputs to this process are:

– the neighbouring partitions mbAddrN\mbPartIdxN\subMbPartIdxN (with N being replaced by A, B, or C),

– the motion vectors mvLXN (with N being replaced by A, B, or C) of the neighbouring partitions,

– the reference indices refIdxLXN (with N being replaced by A, B, or C) of the neighbouring partitions,,

– the reference index refIdxLX of the current partition.

Output of this process is the motion vector prediction mvpLX.

When either partition mbAddrN\mbPartIdxN\subMbPartIdxN is not available or refIdxLXN is not equal to refIdxLX, mvLXN is derived as specified by the following ordered steps:

1. The inverse macroblock scanning process as specified in subclause 6.4.1is invoked with CurrMbAddr as the input and the output is assigned to ( x1, y1 ).

2. The inverse macroblock partition scanning process specified in subclause 6.4.2.1 is invoked with mbPartIdx as the input and the output assigned to ( dx1, dy1 ).

3. The inverse sub-macroblock partition scanning process specified in subclause 6.4.2.2 is invoked with mbPartIdx and subMbPartIdx as the input and the output assigned to ( dx2, dy2 ).

4.  The process specified in subclause J.8.3.1.10 is invoked with depthPic set to DepthCurrPic, dbx1 set to x1 + dx1 + dx2, dby1 set to y1 + dy1 + dy2 and listSuffixFlag as input and InterViewPic, an offset vector dv and an variable InterViewAvailable as outputs.

5. The refIdxCorrespond and mvCorrespond are set as follows.

– If InterViewAvailable is equal to 0, refIdxCorrespond is set to -1, and mvCorrespond [ 0 ] and mvCorrespond [ 1 ] are both set to 0.

– Otherwise, the following step applies in order.

– The variable luma4x4BlkIdx is derived as (4 \* mbPartIdx + subMbPartIdx).

– The inverse 4x4 luma block scanning process as specified in subclause 6.4.3 is invoked with luma4x4BlkIdx as the input and ( x, y ) as the output. In addition, ( xCorrespond, yCorrespond ) is set equal to ( x + ( dv[ 0 ]>>4 ), y + ( dv [ 1 ]>>4 ) ) and mbAddrCorrespond is set equal to ( ( CurrMbAddr / PicWidthInMbs ) + ( dv[1] >>6 ) ) \* PicWidthInMbs + ( CurrMbAddr % PicWidthInMbs ) + ( dv[0] >>6 ).

– Set mbTypeCorrespond to the syntax element mb\_type of the macroblock with address mbAddrCorrespond inside the picture InterViewPic. When mbTypeCorrespond is equal to P\_8x8, P\_8x8ref0, or B\_8x8, subMbTypeCorrespond is set to be the syntax element sub\_mb\_type of the macroblock with address mbAddrCorrespond inside the picture InterViewPic.

– Set mbPartIdxCorrespond to the macroblock partition index of the corresponding partition and subMbPartIdxCorrespond to the sub-macroblock partition index of the corresponding sub-macroblock partition. The derivation process for macroblock and sub-macroblock partition indices as specified in subclause 6.4.12.4 is invoked with the luma location equal to ( xCorrespond, yCorrespond ), the macroblock type equal to mbTypeCorrespond, and when mbTypeCorrespond is equal to P\_8x8, P\_8x8ref0, or B\_8x8, the list of sub-macroblock types subMbTypeCorrespond as the inputs and the outputs are the macroblock partition index mbPartIdxCorrespond and the sub-macroblock partition index subMbPartIdxCorrespond.

– The motion vector mvCorrespond and the reference index refIdxCorrespond are derived as follows.

– If the macroblock mbAddrCorrespond is coded in as Intra prediction mode, both components of mvCorrespond are set equal to 0 and refIdxCorrespond is set equal to –1.

– Otherwise (the macroblock mbAddrCorrespond is not coded as Intra prediction mode), the prediction utilization flags predFlagLXCorrespond is set equal to PredFlagLX[ mbPartIdxCorrespond ], the prediction utilization flag of the macroblock partition mbAddrCorrespond\mbPartIdxCorrespond of the picture InterViewPic. In addition, the following applies.

– When predFlagLXCorrespond is equal to 1, the mvCorrespond and the reference index refIdxCorrespond are set equal to MvLX[ mbPartIdxCorrespond ][ subMbPartIdxCorrespond ] and RefIdxLX[ mbPartIdxCorrespond ], respectively, which are the motion vector mvLX and the reference index refIdxLX that have been assigned to the (sub-)macroblock partition mbAddrCorrespond\mbPartIdxCorrespond\subMbPartIdxCorrespond inside the picture InterViewPic.

6. the motion vectors mvLXN is derived as follows.

– If refIdxCorrespond is equal to refIdxLX,

mvLXN[0] = mvCorrespond[0]

mvLXN[1] = mvCorrespond[1]

– Otherwise,

mvLXN[0] = 0

mvLXN[1] = 0

Each component of the motion vector prediction mvpLX is given by the median of the corresponding vector components of the motion vector mvLXA, mvLXB, and mvLXC:

mvpLX[ 0 ] = Median( mvLXA[ 0 ], mvLXB[ 0 ], mvLXC[ 0 ] ) (J8-XX)

mvpLX[ 1 ] = Median( mvLXA[ 1 ], mvLXB[ 1 ], mvLXC[ 1 ] ) (J8-XX)

J.8.3.1.10 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 InterViewAvailable equal 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.

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 dv is specified as follows:  
index = ViewIdTo3DVAcquisitionParamIndex( view\_id of the current view )  
refIndex = ViewIdTo3DVAcquisitionParamIndex( view\_id of the InterViewPic )

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