J.8.2.1.4 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 interViewAvailable equal to 0.

The following applies to derive an inter-view reference picture or inter-view only reference picture InterViewPic:

for( cIdx = 0;cIdx<=num\_ref\_idx\_l0\_active\_minus1 && !interViewAvailable; cIdx ++)  
 if ( view order index of RefPicList0[ cIdx ] is equal to 0) {  
 InterViewPic = RefPicList0[ cIdx ] (J-34)  
 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 DvAvailable[ i ] and mvCand[ i ] with i equal to 0, 1, and 2 corresponding to neighbouring partitions A, B, and C, respectively, as follows~~

~~for( i = 0; i < 3; i++ )  
 if( view order index of RefPicList0[ refIdxCandL0[ i ] ] is equal to 0 ) {  
 DvAvailable[ i ] = 1  
 mvCand[ i ] = mvCandL0[ i ]  
 } else if( view order index of RefPicList1[ refIdxCandL1[ i ] ] is equal to 0 ) { (J-35)  
 DvAvailable[ i ] = 1  
 mvCand[ i ] = mvCandL1[ i ]  
 } else  
 DvAvailable[ i ] = 0~~

~~– If DvAvailable[ 0 ] + DvAvailable[ 1 ] + DvAvailable[ 2 ] is equal to 1, the following applies:~~

~~dV[ 0 ] = mvCand[ i ][ 0 ]  
dV[ 1 ] = mvCand[ i ][ 1 ] (J-36)~~

~~– Otherwise, the following steps apply in order:~~

– 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 ) ) (J-37)  
 if( depthPic[ dbx1 + i, dby1 + j ] > maxDepth )  
 maxDepth = depthPic[ dbx1 + i, dby1 + j ]

– The variable dispVector is specified as follows:

currIndex = ViewIdTo3DVAcquisitionParamIndex( view\_id of the current view )  
refIndex = ViewIdTo3DVAcquisitionParamIndex( view\_id of the InterViewPic )

dv[ 0 ] = Disparity( maxDepth, currIndex, refIndex )  
dv[ 1 ] = 0

~~dispVector[ 0 ] = Disparity( maxDepth, currIndex, refIndex ) (J-38)  
dispVector[ 1 ] = 0~~

~~– For each value of i equal to 0, 1, and 2, when DvAvailable[ i ] is equal to 0, mvCand[ i ] is set to dispVector.~~

~~– Each component of the variable dV is derived as follows:~~

~~dV[ 0 ] = Median( mvCand[ 0 ][ 0 ], mvCand[ 1 ][ 0 ], mvCand[ 2 ][ 0 ] )  
dV[ 1 ] = Median( mvCand[ 0 ][ 1 ], mvCand[ 1 ][ 1 ], mvCand[ 2 ][ 1 ] ) (J-39)~~