1. * + 1. Derivation process for a disparity sample array

Inputs to this process are:

* a luma location ( xP, yP ) relative to the top-left luma sample of the current picture,
* a disparity vector mvDisp,
* a view order index refViewIdx specifying a reference view,
* a view order index depthViewIdx specifying the view the depth should be derived from
* variables nPSW and nPSH specifying a width and a height, respectively
* a variable splitFlag.

Outputs of this process are:

* a (nPSW)x(nPSH) array disparitySamples of disparities values.

Let refDepPic the picture in the current access unit with ViewIdx( refDepPic ) equal to ViewIdx and DepthFlag( refDepPic ) equal to 1.

Let refDepPels be an array of reconstructed depth samples refDepPic. The luma location (xTL, yTL) of top-left luma sample of a block in refDepPels is derived by

xTL = xP + ( ( mvDisp[ 0 ] + 2 ) >> 2 ) (H‑)

yTL = yP + ( ( mvDisp[ 1 ] + 2 ) >> 2 ) (H‑)

The variables nSubBlkW and nSubBlkW are derived as specified in the following:

* If splitFlag is equal to 1,

The variables xP0, yP0, xP1, yP1, are derived as specified in the following :

xP0 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, xTL)  
yP0 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, yTL)  
xP1 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, xTL + nPSW – 1 )  
yP1 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, yTL + nPSH – 1 )

The variable horSplitFlag is derived as specified in the following:

horSplitFlag = ( refDepPels[ xP0 ][ yP0 ] > refDepPels[ xP1 ][ yP1 ] )  
= = ( refDepPels[ xP1 ][ yP0 ] > refDepPels[ xP0 ][ yP1] ) ) (‑257)

nSubBlkW = horSplitFlag ? 8 : 4 (‑258)

nSubBlkH = horSplitFlag ? 4 : 8 (‑259)

* Otherwise,

nSubBlkW = nPSW (‑260)

nSubBlkH = nPSH (‑261)

The array disparitySamples of size (nPSW)x(nPSH) is derived as specified in the following:

* For sBy in the range of 0 to ( ( nPSH / nSubBlkH) – 1 ), inclusive, the following applies:
  + For sBx in the range of 0 to ( ( nPSW / nSubBlkW) – 1 ), inclusive, the following applies:
    - The variables xB, yB are derived as specified in the following:

xB = sBx \* nSubBlkW  
yB = sBy \* nSubBlkH

* + - The derivation process for a disparity sample block as specified in subclause is invoked with the luma location ( xB, yB ), variables nSubBlkW and nSubBlkH, the array of reconstructed depth samples refDepPels, the luma location ( xTL, yTL ), the view order index refViewIdx, and the array disparitySamples as the inputs, and the output is the modified array disparitySamples

~~The variables nSubBlkW and nSubBlkH are derived as specified in the following:~~

~~SubBlkW = splitFlag ? 8 : nPSW (‑257)~~

~~nSubBlkH = splitFlag ? 8 : nPSH (‑258)~~

~~The array disparitySamples of size (nPSW)x(nPSH) is derived as specified in the following:~~

* ~~For sBy in the range of 0 to ( ( nPSH / nSubBlkH) – 1 ), inclusive, the following applies:~~ 
  + ~~For sBx in the range of 0 to ( ( nPSW / nSubBlkW) – 1 ), inclusive, the following applies:~~ 
    - ~~The variables xB, yB, xP0, yP0, xP1, yP1, are derived as specified in the following:~~

~~xB = sBx \* nSubBlkW  
 yB = sBy \* nSubBlkH  
 xP0 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, x~~~~TL~~ ~~+ xB )  
 yP0 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, y~~~~TL~~ ~~+ yB )  
 xP1 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, x~~~~TL~~ ~~+ xB + nSubBlkW – 1 )  
 yP1 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, y~~~~TL~~ ~~+ yB + nSubBlkH – 1 )~~

* + - ~~The variable nSubSubBlkW is set equal to nSubBlkW and the variable nSubSubBlkH is set equal to nSubBlkH.~~
    - ~~When splitFlag is equal to 1, nSubSubBlkW and nSubSubBlkW are modified as follows:~~ 
      * ~~The variable horSplitFlag is derived as specified in the following:~~ 
        1. ~~horSplitFlag = ( refDepPels[ xP0 ][ yP0 ] > refDepPels[ xP1 ][ yP1 ] )  
            = = ( refDepPels[ xP1 ][ yP0 ] > refDepPels[ xP0 ][ yP1] ) ) (‑259)~~
        2. ~~nSubSubBlkW = horSplitFlag ? nSubSubBlkW : ( nSubSubBlkW >> 1 ) (‑260)~~
        3. ~~nSubSubBlkH = horSplitFlag ? ( nSubSubBlkH >> 1 ) : nSubSubBlkW (‑261)~~
    - ~~The derivation process for a disparity sample block as specified in subclause is invoked with the luma location ( xB, yB ), variables nSubBlkW and nSubBlkH, the array of reconstructed depth samples refDepPels, the luma location ( x~~~~TL~~~~, y~~~~TL~~~~), the variables nSubSubBlkW and nSubSubBlkW, the view order index refViewIdx, and the array disparitySamples as the inputs, and the output is the modified array disparitySamples.~~ 
      * 1. Derivation process for a disparity sample sub block

Inputs to this process are:

* a luma location ( xB, yB ) of the top-left luma sample of the current sub block relative to the top-left luma sample of the current prediction unit,
* variables nSubBlkW and nSubBlkH specifying the width and the height, respectively, of the current sub block.
* an array of reconstructed depth samples refDepPels
* a luma location ( xTL, yTL ) of the top-left luma sample of the current reference block relative to the top-left luma sample of the array of reconstructed depth samples,
* ~~variable nSubSubBlkW and nSubSubBlkW specifying the conversion precision of the corresponding depth samples,~~
* a view order index refViewIdx specifying a reference view,
* an array disparitySamples of disparities values.

Outputs of this process are:

* an modified array disparitySamples of disparities values.

The array disparitySamples is modified as specified in the following:

* The variable maxDep is set equal to –1 and modified as specified in the following.

xP0 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, xTL + xB )  
yP0 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, yTL + yB )   
xP1 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, xTL + xB + nSubBlkW – 1 )  
yP1 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, yTL + yB + nSubBlkH – 1 )

maxDep = Max( maxDep, refDepPels[ xP0 ][ yP0 ] )  
maxDep = Max( maxDep, refDepPels[ xP0 ][ yP1] )  
maxDep = Max( maxDep, refDepPels[ xP1][ yP0 ] )  
maxDep = Max( maxDep, refDepPels[ xP1][ yP1 ] )

* The values of the array depthSamples are modified as specified in the following:

for ( yOff = 0; yOff < nSubBlkH; yOff++ )  
 for( xOff = 0; xOff < nSubBlkW; xOff++ ) {  
 x = xTL + xB + xOff   
 y = yTL + yB + yOff   
 disparitySamples[ x ][ y ] = DepthToDisparityB[ refViewIdx ][ maxDep ]  
 }

* ~~For sBy in the range of 0 to ( ( nSubBlkH / nSubSubBlkH) –1 ), inclusive, the following applies:~~ 
  + ~~For sBx in the range of 0 to ( ( nSubBlkW / nSubSubBlkW) –1 ), inclusive, the following applies:~~ 
    - ~~The variable maxDep is set equal to –1 and modified as specified in the following.~~

~~xSubB = xB + sBx \* nSubSubBlkW  
 ySubB = yB + sBy \* nSubSubBlkH  
 xP0 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, x~~~~TL~~ ~~+ xSubB )  
 yP0 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, y~~~~TL~~ ~~+ ySubB )   
 xP1 = Clip3( 0, pic\_width\_in\_luma\_samples – 1, x~~~~TL~~ ~~+ xSubB + nSubSubBlkW – 1 )  
 yP1 = Clip3( 0, pic\_height\_in\_luma\_samples – 1, y~~~~TL~~ ~~+ ySubB + nSubSubBlkH – 1 )  
 maxDep = Max( maxDep, refDepPels[ xP0 ][ yP0 ] )  
 maxDep = Max( maxDep, refDepPels[ xP0 ][ yP1] )  
 maxDep = Max( maxDep, refDepPels[ xP1][ yP0 ] )  
 maxDep = Max( maxDep, refDepPels[ xP1][ yP1 ] )~~

* + - ~~The values of the array depthSamples are modified as specified in the following:~~

~~for ( yOff = 0; yOff < nSubSubBlkH; yOff++ )  
 for( xOff = 0; xOff < nSubSubBlkW; xOff++ ) {  
 x = xSubB + xOff   
 y = ySubB + yOff   
 disparitySamples[ x ][ y ] = DepthToDisparityB[ refViewIdx ][ maxDep ]  
 }~~