H.7.4.7.1.1 Slice segment header extension semantics

**cp\_scale**[ i ], **cp\_off**[ i ], **cp\_inv\_scale\_plus\_scale**[ i ], and **cp\_inv\_off\_plus\_off**[ i ]specify conversion parameters for converting a depth value to a disparity value. When not present, the values of cp\_scale[ i ], cp\_off[ i ], cp\_inv\_scale\_plus\_scale[ i ], and cp\_inv\_off\_plus\_off[ i ], are inferred to be equal to vps\_cp\_scale[ ViewIdx ][ i ], vps\_cp\_off[ ViewIdx ][ i ], vps\_cp\_inv\_scale\_plus\_scale[ ViewIdx ][ i ], and vps\_cp\_inv\_off\_plus\_off[ ViewIdx ][ i ], respectively, with ViewIdx equal to ViewOrderIndex[ nuh\_layer\_id ].

The variable DDDInvScale[i] is set equal to

(1 << ((BitDepthY  + cp\_precision + 1 ) << 1 )) / cp\_scale[ i ].

The variable DDDShift[i] is set equal to BitDepthY + cp\_precision + 3.

The variable DDDInvOffset[i] is set equal to

-(( DDDInvScale[i] \* cp\_off[ i ] + (Int)(1 << (cp\_precision - 2 ))) >> (cp\_precision - 1 )) + ( 1 << (DDDShift[i] – 1).

…

H.8.5.3.2.1 Derivation process for luma motion vectors for merge mode

…

Outputs of this process are

…

* the flag vspModeFlag, specifying, whether the current PU is coded using view synthesis prediction,
* the flag DDDModeFlag, specifying whether the current PU is coded using disparity derived depth,
* the value disp2depthValue, specifying the disparity derived depth value for the current PU.

…

7. Depending on DepthFlag, the following applies.

mergeCandIsDDDFlag[i] is set equal to 0 for i from 0 to 5 + iv\_mv\_pred\_flag [ nuh\_layer\_id ] + DepthFlag.

* + - If DepthFlag is equal to 0, the variable availableFlagT is set equal to 0.
    - Otherwise ( DepthFlag is equal to 1), the derivation process for the texture merging candidate as specified in subclause is invoked with the luma location ( xPb, yPb ), the variables nPbW and nPbH, the offsets ( nPSW – 1 ) >> 1 and ( nPSH – 1 ) >> 1 and bUseDDD equal to 1 as the inputs and the outputs are the flag availableFlagT, the flag availableFlagD, the disparity derived depth value disp2depth, the prediction utilization flags predFlagL0T and predFlagL1T, predFlagL0D and predFlagL1D, the reference indices refIdxL0T and refIdxL1T, and refIdxL0D, refIdxL1D, and the motion vectors mvL0T and mvL1T and mvL0D and mvL1D.

…

8. The merge candidate lists mergeCandList and mergeCandIsVspFlag are constructed as specified by the following ordered steps:

* 1. The variable numMergeCand is set equal to 0.
  2. When availableFlagT is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to T, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.
  3. When availableFlagT is equal to 1 and availableFlagD is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to D, the entry mergeCandIsVspFlag [ numMergeCand ] is set equal to 0, the entry mergeCandIsDDDFlag[ numMergeCand ] is set equal to 1 and the variable numMergeCand is increased by 1.
  4. When availableFlagIvMC is equal to 1, the entry mergeCandList[ numMergeCand ] is set equal to IvMC, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.

…

1. When availableFlagCol is equal to 1 and numMergeCand is less than 5 + iv\_mv\_pred\_flag[ nuh\_layer\_id ] + DepthFlag, the entry mergeCandList[ numMergeCand ] is set equal to Col, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.
2. If DepthFlag is equal to 0 or numMergeCand >= 5, the variable availableFlagTA is set equal to 0.

Otherwise ( DepthFlag is equal to 1 and numMergeCand < 5), the derivation process for the texture merging candidate as specified in subclause is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, the offsets nPSW and nPSH, and the bUseDDD equal to 0 as the inputs and the outputs are the flag availableFlagTA, the prediction utilization flags predFlagL0TA and predFlagL1TA, the reference indices refIdxL0TA and refIdxL1TA, and the motion vectors mvL0TA and mvL1TA. If there is a mergeCandList[k] equalling to N with k < numMergeCand satisfying, predFlagL0TA==predFlagL0N&& predFlagL1TA ==predFlagL1N&& refIdxL0TA == refIdxL0N&& refIdxL1TA == refIdxL1N&& mvL0TA == mvL0N && mvL1TA == mvL1N, then predFlagL0TA, predFlagL1TA, mvL0TA, and mvL1TA are all set equal to 0; refIdxL0TA and refIdxL1TA are both set equal to -1. Otherwise, mergeCandList[ numMergeCand ] is set equal to TA, the entry mergeCandIsVspFlag[ numMergeCand ] is set equal to 0 and the variable numMergeCand is increased by 1.

…

18. The variable DDDModeFlag is set equal to mergeCandIsDDDFlag[ merge\_idx[ xP][ yP ] ]. When DDDModeFlag is equal to 1, the variable disp2depthValue is set equal to disp2depth.

H.8.5.3.2.14 Derivation process for the texture merging candidate

This process is not invoked when DepthFlag is equal to 0.

Inputs to this process are:

* a luma location ( xPb, yPb ) of the top-left luma sample of the current prediction unit relative to the top-left luma sample of the current picture,
* variables nPbW and nPbH specifying the width and the height, respectively, of the current prediction unit,
* A flag bUseDDD specifying whether to derive the DDD candidate.
* variables nOffsetX and nOffsetY specifying the offset position to get the texture location.

Outputs of this process are:

…

* the motion vectors mvL0T and mvL1T (when availableFlagT is equal to 1),
* a flag availableFlagD specifying whether the disparity derived candidate is available,
* the prediction utilization flags predFlagL0D and predFlagL1D,
* the reference indices refIdxL0D and refIdxL1D (when availableFlagD is equal to 1),
* the motion vectors mvL0D and mvL1D (when availableFlagD is equal to 1),
* the disparity derived depth value disp2depth.
* The variable availableFlagT is set equal to 0. The variables predFlagL0T and predFlagL1T are set equal to 0. The variables refIdxL0T and refIdxL1T are set equal to −1. Both components of the motion vectors mvL0T and mvL1T are set equal to 0.
* The variable availableFlagD is set equal to 0. The variables predFlagL0D and predFlagL1D are set equal to 0. The variables refIdxL0D and refIdxL1D are set equal to −1. Both components of the motion vectors mvL0D and mvL1D are set equal to 0. disp2depth is set equal to 0.

…

The texture luma location ( xRef, yRef ) is derived by:

* 1. xRefFull = xPb + nOffsetX (H‑)
  2. yRefFull = yPb + nOffsetY (H‑)

…

1. When X is equal to 0 or the current slice is a B slice, the following applies:
   * + When textPredFlagLX[ xRef ][ yRef ] is equal to 1, the following applies:
       1. mvLXT[ 0 ] = ( textMvLX[ xRef ][ yRef ][ 0 ] + 2 ) >> 2 (H‑)
       2. mvLXT[ 1 ] = ( textMvLX[ xRef ][ yRef ][ 1 ] + 2 ) >> 2 (H‑)
       3. refIdxLX = textRefIdxLX[ xRef ][ yRef ] (H‑)
       4. predFlagLXT = 1 (H‑)
       5. availableFlagT = 1 (H‑)
          1. When bUseDDD is equal to 1, PicOrderCnt( RefPicListX[ refIdxLX] ) is equal to PicOrderCnt of the current picture and availableFlagD is equal to 0, the following applies
          2. mvLXD[ 0 ] = mvLXT[ 0 ]
          3. mvLXD[ 1 ] = mvLXT[ 1 ]
          4. refIdxLXD = refIdxLX
          5. predFlagLXD = 1
          6. availableFlagD = 1
          7. ViewId = ViewId( RefPicListX[ refIdxLX] ), DV= textMvLX[ xRef ][ yRef ][ 0 ]
          8. disp2depthTmp =
          9. (DDDInvScale[ViewId] \* DV + DDDInvOffset[ViewId])>> DDDShift[ViewId]
          10. disp2depth = Clip3( 0, ( 1 << bitDepth ) − 1, disp2depthTmp)

8.5.3.3 Decoding process for inter prediction samples

…

The variable nCSL is set equal to nCS and the variable nCSC is set equal to nCS >> 1.

* If DDDModeFlag[ xC + xB ][ yC + yB ] is equal to 1, the following applies:

For each luma sample location ( xL = 0..nPbW−1, yL = 0..nPbH−1 ) inside the prediction luma sample array predSamplesL, the corresponding prediction luma sample value predSamplesL[ xL ][ yL ] is set equal to disp2depthValue[ xC + xB ][ yC + yB ]

Else

…