* + - 1. General Coding unit syntax

|  |  |
| --- | --- |
| ~~if ( resPredEnableFlag )~~ |  |
| **~~res\_pred\_flag~~** | ~~ae(v)~~ |

* + - 1. General coding unit semantics

**~~res\_pred\_flag~~** ~~equal to 0 specifies that residual prediction is not used. res\_pred\_flag equal to 1 specifies that residual prediction is used. When res\_pred\_flag is not present, its value shall be inferred to be equal to 0.~~

~~[Ed. (GT). The following paragraph needs to be specified more precisely. ]~~

~~The variable residualCbfNonZero is derived by values of cbf\_luma, cbf\_cb, cbf\_cr and PredMode of the corresponding blocks as follows: residualCbfNonZero is set to 1 if at least one of the corresponding blocks has both PredMode not equal to MODE\_INTRA and any of the values of cbf\_luma, cbf\_cb and cbf\_cr not equal to 0; otherwise, residualCbfNonZero is set equal to 0. The corresponding blocks are identified by the current PU and the disparity vector. All the corresponding blocks belong to transform units that are covered or partially covered by a corresponding rectangle area (of the current PU) in the inter-view reference view component, after shifting the PU location with a disparity vector.~~

~~The variable resPredEnableFlag specifying whether res\_pred\_flag is present in the bitstream is derived as~~

* 1. ~~resPredEnableFlag = multi\_view\_residual\_pred\_flag && residualCbfNonZero && anyTempRefPicFlag (G‑)~~
     + 1. Derivation process for motion vector components and reference indices

….

Let ( xP, yP ) specify the top-left sample location of the current luma prediction block relative to the top-left luma sample of the current picture where xP = xC + xB and yP = yC + yB.

The variable resPredFlag[xP][yP] is initialized as 0.

Let the variable currPic and ListX be the current picture and RefPicListX (with X being 0 or 1) of the current picture, respectively.

…..

* + - 1. Derivation process for inter-view merge candidates

…

The temporal inter-view motion vector merging candidate is derived by the following ordered steps.

* 1. For the prediction list indication X being 0 and 1 the following applies.
     + The derivation process for a temporal inter-view motion vector merging candidate as specified in subclause G.8.5.2.1.15 is invoked with the luma location ( xP, yP ), the variables nPSW and nPSH, the disparity vector mvDisp, the prediction list indication X and the view identifier refViewIdx as the inputs and the outputs are the flag availableFlagLXIvMC, the motion vector mvLXIvMC and the reference index refIdxLXIvMC.
  2. The availability flag availableFlagIvMC, and the prediction utilization flags predFlagL0IvMC and predFlagL1IvMC are derived by
     + 1. availableFlagIvMC = availableFlagL0IvMC | | availableFlagL1IvMC (G‑)  
          predFlagL0IvMC = availableFlagL0IvMC (G‑)  
          predFlagL1IvMC = availableFlagL1IvMC (G‑)
  3. If availableFlagIvMC  is equal to 1, and the PartMode[ xP ][ yP ] is equal to PART\_2Nx2N, then the variable resPredFlag[xP][yP] is set to 1.

The disparity inter-view motion vector merging candidate is derived by the following ordered steps.

…

* + - 1. Decoding process for inter prediction samples

…

When resPredFlag[xB, yB] is equal to 1, the inter-view residual prediction process as specified in subclause is invoked with the luma locations ( xC, yC ) and ( xB, yB ), the size of the current luma coding block nCS, the variables nPSW and nPSH, the reference view index refViewIdx set equal to 0, the prediction list utilization flags, predFlagL0 and predFlagL1, the reference indices refIdxL0 and refIdxL1, and the arrays predSamplesL, predSamplesCb, and predSamplesCr as the inputs and the outputs are modified versions of the arrays predSamplesL, predSamplesCb, and predSamplesCr.

* + - 1. Inter-view residual prediction process

…

Let refResSamplesL be the (PicWidthInSamplesL)x(PicHeightInSamplesL) array of constructed luma residual samples for inter-coded coding units for the view component with ViewIdx equal to refViewIdx. Let refResSamplesCb and refResSamplesCr be the (PicWidthInSamplesL / 2)x(PicHeightInSamplesL / 2) arrays of constructed Cb and Cr residual samples, respectively, for inter-coded coding units for the view component with ViewIdx equal to refViewIdx.

When the flag availableDV is equal to 0 the whole decoding process of this sub-clause terminates.

~~The variable log2resPredDenom is set equal to 0 and the following ordered steps apply.~~

* 1. ~~When predFlagL0 is equal to 1 and ViewIdx is not equal to the view order index of RefPicListL0[ refIdxL0 ], log2resPredDenom is set equal to log2resPredDenom + 1.~~
  2. ~~When predFlagL1 is equal to 1 and ViewIdx is not equal to the view order index of RefPicListL1[ refIdxL1 ], log2resPredDenom is set equal to log2resPredDenom + 1.~~
  3. ~~The variable log2MaxResPredDenom is derived by~~
     1. ~~log2MaxResPredDenom = ( ( predFlagL0 = = 1 ) && ( predFlagL1 == 1 ) ? 1 : 0 ) (G‑188)~~

~~When log2resPredDenom is greater than log2MaxResPredDenom the whole decoding process of this sub-clause terminates.~~

For y proceeding over the values 0..(nPSH – 1) and x proceeding over the values 0..(nPSW – 1), the following ordered steps apply.

* 1. The variables xR is derived by
     + 1. xR0 = Clip3( 0, PicWidthInSamplesL – 1, xP + x + ((mvDisp[ 0 ]+2) >> 2 ) ) (G‑189)
  2. The sample predSamplesL[ x, y ] is modified by
     + 1. deltaL = refResSamplesL[ xR, y ] (G‑193)  
          predSamplesL[ x, y ] = predSamplesL[ x, y ] + deltaL (G‑194)

For y proceeding over the values 0..(nPSH / 2 – 1) and x proceeding over the values 0..(nPSW / 2 – 1), the following ordered steps are specified:

* 1. The variables xR is derived by
     + 1. xR = Clip3( 0, PicWidthInSamplesL / 2 – 1, xP / 2 + x + ((mvDisp[0]+4) >> 3 ) ) (G‑195)
  2. The sample predSamplesCb[ x, y ] is modified by
     + 1. deltaCb = refResSamplesCb[ xR, y ] (G‑199)  
          predSamplesCb[ x, y ] = predSamplesCb[ x, y ] + deltaCb  (G‑200)
  3. The sample predSamplesCr[ x, y ] is modified by
     + 1. deltaCr = refResSamplesCr[ xR, y ] (G‑201)  
          predSamplesCr[ x, y ] = predSamplesCr[ x, y ] + deltaCr  (G‑202)