|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  4th Meeting: Incheon, KR, 20–26 Apr. 2013 | Document: JCT3V-D0118 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Proposed text for JCT3V-D0118** | | |
| *Status:* | Input document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Jin Young Lee  Chanyul Kim | Email: | [jinyoung79.lee@samsung.com](mailto:jinyoung79.lee@samsung.com) |
| *Source:* | Samsung Electronics Co. Ltd. | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

H.8.5.2.1 Derivation process for motion vector components and reference indices

Input to this process are

* a luma location ( xC, yC ) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
* a luma location ( xB, yB ) of the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block,
* a variable nCS specifying the size of the current luma coding block,
* variables specifying the width and the height of the luma prediction block, nPbW and nPbH,
* a variable partIdx specifying the index of the current prediction unit within the current coding unit.

Outputs of this process are

* luma motion vectors mvL0 and mvL1
* chroma motion vectors mvCL0 and mvCL1,
* reference indices refIdxL0 and refIdxL1,
* prediction list utilization flags predFlagL0 and predFlagL1.

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.

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

The function LongTermRefPic( picX, refIdx, LX ), with X being either 0 or 1, is defined as follows. If the picture with index refIdx from reference picture list LX of the picture picX was marked as "used for long term reference" at the time when picX was the current picture, LongTermRefPic( picX, refIdx, LX ) is equal to 1; otherwise LongTermRefPic( picX, refIdx, LX ) is equal to 0.

The flag vspModeFlag is set equal to 0, the variable refViewIdx is set equal to −1 and the flag availableFlagIvMC is set equal to 0.

For X being replaced by 0 and 1, the variable ivpMvFlagLX and the variable ivpMvDispLX are derived as:

* 1. ivpMvFlagLX = 0 (H‑68)  
     ivpMvDispLX[ 0 ] = 0 (H‑69)  
     ivpMvDispLX[ 1 ] = 0 (H‑70)

For the derivation of the variables mvL0 and mvL1, refIdxL0 and refIdxL1 as well as prefFlagL0 and prefFlagL1, the following applies.

* If PredMode[ xC ][ yC ] is equal to MODE\_SKIP, the derivation process for luma motion vectors for merge mode as specified in subclause H.8.5.2.1.1 is invoked with the luma location ( xC, yC ), the luma location ( xP, yP ), variables nCS, nPbW, nPbH and the partition index partIdx as inputs and the output being the luma motion vectors mvL0, mvL1, the reference indices refIdxL0, refIdxL1, ~~and~~ the prediction list utilization flags predFlagL0 and predFlagL1, the disparity vector availability flags ivpMvFlagL0 and ivpMvFlagL1, the disparity vectors ivpMvDispL0 and ivpMvDispL1, the flag vspModeFlag, the variable refViewIdx and the flag availableFlagIvMC. .
* Otherwise, if PredMode[ xC ][ yC ] is equal to MODE\_INTER and merge\_flag[ xP ][ yP ] is equal to 1, the derivation process for luma motion vectors for merge mode as specified in subclause H.8.5.2.1.1 is invoked with the luma location ( xC, yC ), luma location ( xP, yP ), variables nCS, nPbW and nPbH and the partition index partIdx as inputs and the outputs being the luma motion vectors mvL0 and mvL1, the reference indices refIdxL0 and refIdxL1, the prediction utilization flags predFlagL0 and predFlagL1, the disparity vector availability flags ivpMvFlagL0 and ivpMvFlagL1, the disparity vectors ivpMvDispL0 and ivpMvDispL1, the flag vspModeFlag, the variable refViewIdx and the flag availableFlagIvMC. .
* Otherwise, for X being replaced by either 0 or 1 in the variables predFlagLX, mvLX, refIdxLX and in Pred\_LX and in the syntax elements ref\_idx\_lX and MvdLX, the following applies.
  + 1. The variables refIdxLX and predFlagLX are derived as follows.
       - If inter\_pred\_idc[ xP ][ yP ] is equal to Pred\_LX or Pred\_BI,

refIdxLX = ref\_idx\_lX[ xP ][ yP ] (H‑71)  
predFlagLX = 1 (H‑72)

* + - * Otherwise, the variables refIdxLX and predFlagLX are specified by

refIdxLX = −1 (H‑73)  
predFlagLX = 0 (H‑74)

* + 1. The variable mvdLX is derived as follows.
       - 1. mvdLX[ 0 ] = MvdLX[ xP ][ yP ][ 0 ] (H‑75)  
            mvdLX[ 1 ] = MvdLX[ xP ][ yP ][ 1 ] (H‑76)
    2. When predFlagLX is equal to 1, the derivation process for luma motion vector prediction in subclause H.8.5.2.1.5 is invoked with the luma coding block location ( xC, yC ), the coding block size nCS, the luma prediction block location ( xP, yP ), variables nPbW and nPbH, refIdxLX, and the partition index partIdx as the inputs and the output being mvpLX.
    3. When predFlagLX is equal to 1, the luma motion vector mvLX is derived as
       - 1. uLX[ 0 ] = (mvpLX[ 0 ] + mvdLX[ 0 ] + 216) % 216 (H‑77)  
            mvLX[ 0 ] = ( uLX[ 0 ] >= 215 ) ? ( uLX[ 0 ] − 216 ) : uLX[ 0 ] (H‑78)  
            uLX[ 1 ] = (mvpLX[ 1 ] + mvdLX[ 1 ] + 216) % 216 (H‑79)  
            mvLX[ 1 ] = ( uLX[ 1 ] >= 215 ) ? ( uLX[ 1 ] − 216 ) : uLX[ 1 ] (H‑80)
         2. NOTE – The resulting values of mvLX[ 0 ] and mvLX[ 1 ] as specified above will always be in the range of −215 to 215 − 1.

When ChromaArrayType is not equal to 0 and predFlagLX (with X being either 0 or 1) is equal to 1, the derivation process for chroma motion vectors in subclause 8.5.2.1.8 is invoked with mvLX and refIdxLX as inputs and the output being mvCLX.

If iv\_res\_pred\_flag[ nuh\_layer\_id ] is equal to 1, availableFlagIvMC is equal to 1, and PartMode[ xP ][ yP ] is equal to PART\_2Nx2N, ResPredFlag[ xP ][ yP ] is set equal to 1, otherwise ( iv\_res\_pred\_flag[ nuh\_layer\_id ] is equal to 0 or availableFlagIvMC is equal to 0 or PartMode[ xP ][ yP ] is not equal to PART\_2Nx2N ), ResPredFlag[ xP ][ yP ] is set equal to 0.

If view\_synthesis\_pred\_flag is equal to 1, availableFlagVSP is equal to 1, and PartMode[ xP ][ yP ] is equal to PART\_2Nx2N, VspModeFlag [ xP ][ yP ] is set equal to 1, otherwise (view\_synthesis\_pred\_flag is equal to 0 or availableFlagVSP is equal to 0 or PartMode[ xP ][ yP ] is not equal to PART\_2Nx2N ), VspModeFlag [ xP ][ yP ] is set equal to 0.

For use in derivation processes of variables invoked later in the decoding process, the following assignments are made for x = xP.. ( xP + nPbW − 1 ), y = yP..( yP + nPbH− 1 ) (with X being either 0 or 1):

* 1. IvpMvFlagLX[ x ][ y ] = ivpMvFlagLX (H‑81)  
     IvpMvDispLX[ x ][ y ] = ivpMvDispLX (H‑82)  
       
     RefViewIdx[ x ][ y ] = refViewIdx (H‑84)