|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  14th Meeting: Vienna, AT, 25 July – 2 Aug. 2013 | Document: JCTVC-N0185-r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **On low-delay checking process for SHVC** | | |
| *Status:* | Input Document | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Xiaoyu Xiu, Yan Ye, Yuwen He, Yong He  9710 Scranton R-D, #250 San Diego, CA 92121 USA  Yongbing Lin, Xiaozhen Zheng, Xu Chen, Jianhua Zheng  No. 156, Rd. Bei-qing Hai-dian district 100095 Beijing, China | Tel: Email: | 1-858-210-4830 [Xiaoyu.Xiu@InterDigital.com](mailto:Xiaoyu.Xiu@InterDigital.com)  86-10-60612372 [yblin@huawei.com](mailto:yblin@huawei.com) |
| *Source:* | InterDigital Communications, Inc., HiSilicon Technologies | | |

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

# Abstract

In this contribution, the low-delay checking process in SHVC Test Model (SHM2.0) is modified to reportedly improve the efficiency of temporal motion vector prediction (TMVP) for enhancement layer (EL) coding. More specifically, the low-delay flag is set to true if the inter-layer prediction (ILP) picture is used as the co-located picture for EL TMVP derivation, such that the motion vector (MV) of the co-located prediction unit (PU) always comes from the same reference picture list of the target MV of the current PU for better TMVP prediction. The proposed modification is a slice-level change as the low-delay flag is determined per slice and referred to by all the PUs of a coded picture. Experimental results show that the proposed change reportedly achieves 0.4%, 0.4% and 0.5% BD-rate savings on average for 2x, 1.5x and SNR scalability in RA configuration, compared to the anchors of SHM2.0.

# Introduction

The current SHVC solution is built upon the reference index based framework where the ILP picture (up-sampled if necessary) generated from the base layer (BL) is used as one additional reference picture along with the temporal reference pictures to predict the current EL picture [1]. The reference index is signalled to indicate whether the current PU is predicted from the ILP picture, or the temporal reference pictures, or the combination of both. Moreover, the inter-layer motion prediction is enabled by setting the ILP picture as the co-located picture for the EL TMVP derivation. The motion information of the ILP picture is obtained from the motion field of the BL picture based on the motion field mapping process [2, 3].

In reference index based SHVC, the TMVP candidate is derived in the same way as in HEVC [4]. Figure 1 illustrates EL TMVP derivation process in SHVC, when the ILP picture is selected as the co-located picture. Given the mapped motion information of the ILP picture, the MV (*mvCol*) of the co-located PU (*colPU*)is used to predict the MV (*mvLX*) of the current PU *currPU*. If *colPU* is a bi-predicted PU, there could be two MVs of the co-located PU, i.e., the forward MV from the reference list L0 and the backward MV (*mvCol’*) from the reference list L1, available to predict *mvCol*. And, the reference list *listCol* used to derive TMVP candidate is determined based on the target reference list *LX* of *currPU*, the low-delay flag and the reference picture list where the co-located picture is from, i.e., collocated\_from\_l0 \_flag [1, 4]. More specifically, if low-delay flag is equal to true, *listCol* is set equal to *LX*; otherwise, *listCol* is set equal to the reference picture list indicated by collocated\_from\_l0\_flag which is signaled in slice segment header. Note that although *listCol* is assumed to be equal to *LX* in Figure 1, they may represent different reference lists in practice.

In both SHVC and HEVC, the low-delay flag is set to indicate if the POCs of all the reference pictures of the current picture are less than or equal to its own POC. The low-delay flag is set before encoding/decoding a slice and referred to by the TMVP derivation processes of all the PUs in the same slice.

Figure 1. EL TMVP derivation when ILP picture is used as the co-located picture

# Proposed method

As shown in Figure 1, when the ILP picture is selected as the co-located picture for EL TMVP, the temporal reference pictures of the current picture in both lists L0 and L1 have exactly the same POCs as that of the ILP picture. This is true when the BL and the EL have the same prediction structure, which is the common case for most applications. In this case, a better TMVP prediction efficiency could be expected if *currPU* and *colPU* use the same reference list (i.e., *listCol* = *LX*), given the strong correlation between *mvCol* and *mvLX* which come from the same reference picture list. Therefore, in this contribution, it is proposed to set the low-delay flag to be always true when the ILP picture is selected as the co-located picture of EL TMVP derivation process.

The proposed modification is a high-level change given that the low-delay is determined at slice-level and does not change throughout the encoding/decoding of a slice. Therefore, the proposed change does fulfill the spirit of the current design of reference index based SHVC that is targeted at not changing the operations lower than slice-level encoding/decoding of single-layer HEVC.

# Experimental results

The proposed change is implemented based on SHM2.0 and tested under the common test conditions specified in [5]. The results of reference index based SHM2.0 are used as anchor for BD-rate calculation. Both the encoding time and the decoding time are obtained a heterogeneous cluster system.

Given that the low-delay flag remains to be true for AI, LDP and LDB configurations (the POCs of all the reference pictures are always smaller or equal to the POC of the current picture), the proposed modification only influences the performance of RA configuration. Experimental results under RA configuration show that the proposed low-delay flag modification reportedly achieve 0.4%, 0.4% and 0.5% luma BD-rate savings for 2x, 1.5x and SNR scalability, respectively. When considering chroma components, the corresponding average BD-rate savings are about 0.6%, 0.7% and 0.9% respectively.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RA HEVC 2x** | | | **RA HEVC 1.5x** | | | **RA HEVC SNR** | | |
|  | Y | U | V | Y | U | V | Y | U | V |
| Class A | -0.3% | -0.5% | -0.5% |  |  |  | -0.4% | -0.8% | -0.9% |
| Class B | -0.4% | -0.7% | -0.7% | -0.4% | -0.7% | -0.7% | -0.5% | -1.0% | -1.1% |
| **Overall (Test vs Ref)** | -0.4% | -0.6% | -0.7% | -0.4% | -0.7% | -0.7% | -0.5% | -0.9% | -1.0% |
| **Overall (Test vs single layer)** | 18.7% | 32.4% | 31.1% | 15.7% | 28.0% | 28.2% | 13.9% | 30.9% | 32.8% |
| **Overall (Ref vs single layer)** | 19.2% | 33.3% | 32.0% | 16.2% | 28.8% | 29.1% | 14.4% | 32.1% | 34.1% |
| **EL only (Test vs Ref)** | -0.6% | -0.9% | -0.9% | -0.6% | -0.9% | -1.0% | -0.7% | -1.2% | -1.3% |
| Enc Time[%] | 111.5% | | | 116.9% | | | 120.4% | | |
| Dec Time[%] | 107.7% | | | 115.7% | | | 119.7% | | |
| Enc Mem[%] | #DIV/0! | | | #DIV/0! | | | #DIV/0! | | |
| BL Match | Matched | | | Matched | | | Matched | | |

# Conclusion

In this contribution, the low-delay checking process in reference index based SHVC is modified by setting the low-delay flag to true if ILP picture is used as the co-located picture for EL TMVP derivation. The proposed modification of the low-delay flag is carried out at the slice-level and referred to by the TMVP process of all the PUs in the same slice. Experimental results show that the proposed change reportedly achieves 0.4%, 0.4% and 0.5% BD-rate savings on average for 2x, 1.5x and SNR scalability in RA configuration, compared to SHM2.0 anchors.

# Patent rights declaration(s)

**InterDigital Communications Inc. may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**HiSilicon Technologies and Huawei Technologies may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

# References

1. J. Chen, J. Boyce, Y. Ye and M. M. Hannuksela, “SHVC Working Draft 2”, JCTVC document JCTVC-M1008, Incheon, Korea, April, 2013.
2. X. Xiu, Y. He, Y. He and Y. Ye, “TE C5: Motion field mapping”, JCTVC document JCTVC-L0052, Geneva, Switzerland, January, 2013.
3. J. Chen, V. Seregin, L. Guo and M. Karczewicz, “Non-TE5: on motion mapping in SHVC”, JCTVC document JCTVC-L0336, Geneva, Switzerland, January, 2013.
4. B. Bross, W-J. Han, J-R. Ohm, G. J. Sullivan, High Efficiency Video Coding (HEVC) Text Specification Draft 10. JCTVC document JCTVC-L1003, Geneva, Switzerland, January, 2013.
5. X. Li, J. Boyce, P. Onno and Y. Ye, “Common Test Conditions and Software Reference Configurations for the Scalable Test Model”, JCTVC document JCTVC-M1009, Incheon, Korea, April, 2013.

# Appendix. Proposed text changes

The specification text changes of the proposed low-delay flag modification are provided in this section. All the changes are made based on SHVC working draft 2[1] and HEVC text specification draft 10 [4], with newly added text highlighted in yellow. There are three different methods of incorporating the proposed change into the current SHVC working draft.

Method one:

G.8.5.3.2.7 Derivation process for temporal luma motion vector prediction

Inputs to this process are:

* a luma location ( xPb, yPb ) specifying the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture,
* two variables nPbW and nPbH specifying the width and the height of the luma prediction block,
* a reference index refIdxLX, with X being 0 or 1.

Outputs of this process are:

* the motion vector prediction mvLXCol,
* the availability flag availableFlagLXCol.

The variable currPb specifies the current luma prediction block at luma location ( xPb, yPb ).

The variables mvLXCol and availableFlagLXCol are derived as follows:

* If slice\_temporal\_mvp\_enabled\_flag is equal to 0, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
* Otherwise, the following ordered steps apply:

1. Depending on the values of alt\_collocated\_indication\_flag, collocated\_ref\_layer\_idx, slice\_type, collocated\_from\_l0\_flag, and collocated\_ref\_idx, the variable colPic, specifying the collocated picture, is derived as follows:

* If alt\_collocated\_indication\_flag is equal to 1, colPic is set equal to the picture in the current access unit with nuh\_layer\_id equal to ActiveMotionPredRefLayerId[ collocated\_ref\_layer\_idx ].
* Otherwise, if slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 0, colPic is set equal to RefPicList1[ collocated\_ref\_idx ].
* Otherwise (slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 1 or slice\_type is equal to P), colPic is set equal to RefPicList0[ collocated\_ref\_idx ].

1. If DiffPicOrderCnt( colPic, currPic ) is equal to 0, the value of collocate\_from\_l0\_flag is set to X, with X indicating the list represented by refIdxLX.
2. The bottom right collocated motion vector is derived as follows:

xColBr = xPb + nPbW (G‑57)

yColBr = yPb + nPbH (G‑58)

* If yPb  >>  CtbLog2SizeY is equal to yColBr  >>  CtbLog2SizeY, yColBr is less than pic\_height\_in\_luma\_samples, and xColBr is less than pic\_width\_in\_luma\_samples, the following applies:
* The variable colPb specifies the luma prediction block covering the modified location given by ( ( xColBr  >>  4 )  <<  4, ( yColBr  >>  4 )  <<  4 ) inside the collocated picture specified by colPic.
* The luma location ( xColPb, yColPb ) is set equal to the top-left sample of the collocated luma prediction block specified by colPb relative to the top-left luma sample of the collocated picture specified by colPic.
* The derivation process for collocated motion vectors as specified in subclause 8.5.3.2.8 is invoked with currPb, colPic, colPb, ( xColPb, yColPb ), and refIdxLX as inputs, and the output is assigned to mvLXCol and availableFlagLXCol.
* Otherwise, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.

1. When availableFlagLXCol is equal to 0, the central collocated motion vector is derived as follows:

xColCtr = xPb + ( nPbW  >>  1 ) (G‑59)

yColCtr = yPb + ( nPbH  >>  1 ) (G‑60)

* The variable colPb specifies the luma prediction block covering the modified location given by ( ( xColCtr  >>  4 )  <<  4, ( yColCtr  >>  4 )  <<  4 ) inside the colPic.
* The luma location ( xColPb, yColPb ) is set equal to the top-left sample of the collocated luma prediction block specified by colPb relative to the top-left luma sample of the collocated picture specified by colPic.
* The derivation process for collocated motion vectors as specified in subclause 8.5.3.2.8 is invoked with currPb, colPic, colPb, ( xColPb, yColPb ), and refIdxLX as inputs, and the output is assigned to mvLXCol and availableFlagLXCol.

Method two:

The second method enables the proposed low-delay checking process by following the same way of current SHM2.0 implementation. More specially, the method firstly sets the low-delay flag prior to the decoding process of each slice. Then, the low-delay flag is referred to by the TMVP derivation process when decoding the blocks in that slice.

## 8.1 General decoding process

Input to this process is a bitstream. Output of this process is a list of decoded pictures.

The layer identifier list TargetDecLayerIdList, which specifies the list of nuh\_layer\_id values, in increasing order of nuh\_layer\_id values, of the NAL units to be decoded, is specified as follows:

– If some external means, not specified in this Specification, is available to set TargetDecLayerIdList, TargetDecLayerIdList is set by the external means.

– Otherwise, if the decoding process is invoked in a bitstream conformance test as specified in subclause , TargetDecLayerIdList is set as specified in subclause .

– Otherwise, TargetDecLayerIdList contains only one nuh\_layer\_id value that is equal to 0.

The variable HighestTid, which identifies the highest temporal sub-layer to be decoded, is specified as follows:

– If some external means, not specified in this Specification, is available to set HighestTid, HighestTid is set by the external means.

– Otherwise, if the decoding process is invoked in a bitstream conformance test as specified in subclause , HighestTid is set as specified in subclause .

– Otherwise, HighestTid is set equal to sps\_max\_sub\_layers\_minus1.

The sub-bitstream extraction process as specified in clause is applied with the bitstream, HighestTid, and TargetDecLayerIdList as inputs, and the output is assigned to a bitstream referred to as BitstreamToDecode.

The decoding processes specified in the remainder of this subclause apply to each coded picture, referred to as the current picture and denoted by the variable CurrPic, in BitstreamToDecode.

Depending on the value of chroma\_format\_idc, the number of sample arrays of the current picture is as follows:

– If chroma\_format\_idc is equal to 0, the current picture consists of 1 sample array SL.

– Otherwise (chroma\_format\_idc is not equal to 0), the current picture consists of 3 sample arrays SL, SCb, SCr.

The decoding process for the current picture takes as inputs the syntax elements and upper-case variables from clause . When interpreting the semantics of each syntax element in each NAL unit, the term "the bitstream" (or part thereof, e.g. a CVS of the bitstream) refers to BitstreamToDecode (or part thereof).

The decoding process is specified such that all decoders will produce numerically identical cropped decoded pictures. Any decoding process that produces identical cropped decoded pictures to those produced by the process described herein (with the correct output order or output timing, as specified) conforms to the decoding process requirements of this Specification.

When the current picture is a BLA picture that has nal\_unit\_type equal to BLA\_W\_LP or is a CRA picture, the following applies:

– If some external means not specified in this Specification is available to set the variable UseAltCpbParamsFlag to a value, UseAltCpbParamsFlag is set equal to the value provided by the external means.

– Otherwise, the value of UseAltCpbParamsFlag is set equal to 0.

When the current picture is an IRAP picture, the following applies:

– If the current picture is an IDR picture, a BLA picture, the first picture in the bitstream in decoding order, or the first picture that follows an end of sequence NAL unit in decoding order, the variable NoRaslOutputFlag is set equal to 1.

– Otherwise, if some external means not specified in this Specification is available to set the variable HandleCraAsBlaFlag to a value for the current picture, the variable HandleCraAsBlaFlag is set equal to the value provided by the external means and the variable NoRaslOutputFlag is set equal to HandleCraAsBlaFlag.

– Otherwise, the variable HandleCraAsBlaFlag is set equal to 0 and the variable NoRaslOutputFlag is set equal to 0.

Depending on the value of separate\_colour\_plane\_flag, the decoding process is structured as follows:

– If separate\_colour\_plane\_flag is equal to 0, the decoding process is invoked a single time with the current picture being the output.

– Otherwise (separate\_colour\_plane\_flag is equal to 1), the decoding process is invoked three times. Inputs to the decoding process are all NAL units of the coded picture with identical value of colour\_plane\_id. The decoding process of NAL units with a particular value of colour\_plane\_id is specified as if only a CVS with monochrome colour format with that particular value of colour\_plane\_id would be present in the bitstream. The output of each of the three decoding processes is assigned to one of the 3 sample arrays of the current picture, with the NAL units with colour\_plane\_id equal to 0, 1, and 2 being assigned to SL, SCb, and SCr, respectively.

NOTE – The variable ChromaArrayType is derived as equal to 0 when separate\_colour\_plane\_flag is equal to 1 and chroma\_format\_idc is equal to 3. In the decoding process, the value of this variable is evaluated resulting in operations identical to that of monochrome pictures (when chroma\_format\_idc is equal to 0).

The decoding process operates as follows for the current picture CurrPic:

1. The decoding of NAL units is specified in subclause .
2. The processes in subclause  specify the following decoding processes using syntax elements in the slice segment layer and above:

– Variables and functions relating to picture order count are derived in subclause . This needs to be invoked only for the first slice segment of a picture.

– The decoding process for RPS in subclause  is invoked, wherein reference pictures may be marked as "unused for reference" or "used for long-term reference". This needs to be invoked only for the first slice segment of a picture.

– When the current picture is a BLA picture or is a CRA picture with NoRaslOutputFlag equal to 1, the decoding process for generating unavailable reference pictures specified in subclause  is invoked, which needs to be invoked only for the first slice segment of a picture.

– PicOutputFlag is set as follows:

– If the current picture is a RASL picture and NoRaslOutputFlag of the associated IRAP picture is equal to 1, PicOutputFlag is set equal to 0.

– Otherwise, PicOutputFlag is set equal to pic\_output\_flag.

– At the beginning of the decoding process for each P or B slice, the decoding process for reference picture lists construction specified in subclause  is invoked for derivation of reference picture list 0 (RefPicList0) and, when decoding a B slice, reference picture list 1 (RefPicList1).

– The process as specified in subclause 8.3.5 is invoked to set the variables colPic and LowDelayFlag.

1. The processes in subclauses , , , and specify decoding processes using syntax elements in all syntax structure layers. It is a requirement of bitstream conformance that the coded slices of the picture shall contain slice segment data for every coding tree unit of the picture, such that the division of the picture into slices, the division of the slices into slice segments, and the division of the slice segments into coding tree units each forms a partitioning of the picture.
2. After all slices of the current picture have been decoded, the decoded picture is marked as "used for short-term reference".

### 8.3.5 Decoding process for setting collocated picture and low-delay flag

Inputs to this process are reference picture list RefPicList0 and RefPicList1 when decoding a B slice.

Output of this process are:

* The variable colPic specifying the collocated picture of the current slice.
* The variable LowDelayFlag.

This process is invoked per slice, after the decoding process of reference picture list construction but prior to decoding any coding unit.

Depending on the values of alt\_collocated\_indication\_flag, collocated\_ref\_layer\_idx, slice\_type, collocated\_from\_l0\_flag, and collocated\_ref\_idx, the variable colPic, specifying the collocated picture, is derived as follows:

* If alt\_collocated\_indication\_flag is equal to 1, colPic is set equal to the picture in the current access unit with nuh\_layer\_id equal to ActiveMotionPredRefLayerId[ collocated\_ref\_layer\_idx ].
* Otherwise, if slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 0, colPic is set equal to RefPicList1[ collocated\_ref\_idx ].
* Otherwise (slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 1 or slice\_type is equal to P), colPic is set equal to RefPicList0[ collocated\_ref\_idx ].

Depending on the picture order counts of the reference pictures and colPic, the variable LowDelayFlag is derived as follows:

* If DiffPicOrderCnt( colPic, currPic ) is equal to 0, LowDelayFlag is set equal to 1.
* Otherwise, if DiffPicOrderCnt( aPic, currPic ) is less than or equal to 0 for each reference picture aPic in RefPicList0 and RefPicList1 of current slice, LowDelayFlag is set equal to 1.
* Otherwise, LowDelayFlag is set equal to 0.

##### 8.5.3.2.7 Derivation process for temporal luma motion vector prediction

Inputs to this process are:

* a luma location ( xPb, yPb ) specifying the top-left sample of the current luma prediction block relative to the top-left luma sample of the current picture,
* two variables nPbW and nPbH specifying the width and the height of the luma prediction block,
* a reference index refIdxLX, with X being 0 or 1.

Outputs of this process are:

* the motion vector prediction mvLXCol,
* the availability flag availableFlagLXCol.

The variable currPb specifies the current luma prediction block at luma location ( xPb, yPb ).

The variables mvLXCol and availableFlagLXCol are derived as follows:

* If slice\_temporal\_mvp\_enabled\_flag is equal to 0, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
* Otherwise, the following ordered steps apply:

1. ~~Depending on the values of slice\_type, collocated\_from\_l0\_flag, and collocated\_ref\_idx, the variable colPic, specifying the collocated picture, is derived as follows:~~

* ~~If slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 0, colPic is set equal to RefPicList1[ collocated\_ref\_idx ].~~
* ~~Otherwise (slice\_type is equal to B and collocated\_from\_l0\_flag is equal to 1 or slice\_type is equal to P), colPic is set equal to RefPicList0[ collocated\_ref\_idx ].~~

1. The bottom right collocated motion vector is derived as follows:

xColBr = xPb + nPbW (8‑173)

yColBr = yPb + nPbH (8‑174)

* If yPb  >>  CtbLog2SizeY is equal to yColBr  >>  CtbLog2SizeY, yColBr is less than pic\_height\_in\_luma\_samples, and xColBr is less than pic\_width\_in\_luma\_samples, the following applies:
* The variable colPb specifies the luma prediction block covering the modified location given by ( ( xColBr  >>  4 )  <<  4, ( yColBr  >>  4 )  <<  4 ) inside the collocated picture specified by colPic.
* The luma location ( xColPb, yColPb ) is set equal to the top-left sample of the collocated luma prediction block specified by colPb relative to the top-left luma sample of the collocated picture specified by colPic.
* The derivation process for collocated motion vectors as specified in subclause 8.5.3.2.8 is invoked with currPb, colPic, colPb, ( xColPb, yColPb ), and refIdxLX as inputs, and the output is assigned to mvLXCol and availableFlagLXCol.
* Otherwise, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.

1. When availableFlagLXCol is equal to 0, the central collocated motion vector is derived as follows:

xColCtr = xPb + ( nPbW  >>  1 ) (8‑175)

yColCtr = yPb + ( nPbH  >>  1 ) (8‑176)

* The variable colPb specifies the luma prediction block covering the modified location given by ( ( xColCtr  >>  4 )  <<  4, ( yColCtr  >>  4 )  <<  4 ) inside the colPic.
* The luma location ( xColPb, yColPb ) is set equal to the top-left sample of the collocated luma prediction block specified by colPb relative to the top-left luma sample of the collocated picture specified by colPic.

The derivation process for collocated motion vectors as specified in subclause 8.5.3.2.8 is invoked with currPb, colPic, colPb, ( xColPb, yColPb ), and refIdxLX as inputs, and the output is assigned to mvLXCol and availableFlagLXCol.

### 8.5.3.2.8 Derivation process for collocated motion vectors

Inputs to this process are:

* a variable currPb specifying the current prediction block,
* a variable colPic specifying the collocated picture,
* a variable colPb specifying the collocated prediction block inside the collocated picture specified by colPic,
* a luma location ( xColPb, yColPb ) specifying the top-left sample of the collocated luma prediction block specified by colPb relative to the top-left luma sample of the collocated picture specified by colPic,
* a reference index refIdxLX, with X being 0 or 1.

Outputs of this process are:

* the motion vector prediction mvLXCol,
* the availability flag availableFlagLXCol.

The variable currPic specifies the current picture.

The arrays predFlagLXCol[ x ][ y ], mvLXCol[ x ][ y ], and refIdxLXCol[ x ][ y ] are set equal to the corresponding arrays of the collocated picture specified by colPic, PredFlagLX[ x ][ y ], MvLX[ x ][ y ], and RefIdxLX[ x ][ y ], respectively, with X being the value of X this process is invoked for.

The variables mvLXCol and availableFlagLXCol are derived as follows:

* If colPb is coded in an intra prediction mode, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
* Otherwise, the motion vector mvCol, the reference index refIdxCol, and the reference list identifier listCol are derived as follows:
  + If predFlagL0Col[ xColPb ][ yColPb ] is equal to 0, mvCol, refIdxCol, and listCol are set equal to mvL1Col[ xColPb ][ yColPb ], refIdxL1Col[ xColPb ][ yColPb ], and L1, respectively.
  + Otherwise, if predFlagL0Col[ xColPb ][ yColPb ] is equal to 1 and predFlagL1Col[ xColPb ][ yColPb ] is equal to 0, mvCol, refIdxCol, and listCol are set equal to mvL0Col[ xColPb ][ yColPb ], refIdxL0Col[ xColPb ][ yColPb ], and L0, respectively.
  + Otherwise (predFlagL0Col[ xColPb ][ yColPb ] is equal to 1 and predFlagL1Col[ xColPb ][ yColPb ] is equal to 1), the following assignments are made:
    - * ~~If DiffPicOrderCnt( aPic, currPic ) is less than or equal to 0 for every picture aPic in every reference picture list of the current slice~~If LowDelayFlag is equal to 1, mvCol, refIdxCol, and listCol are set equal to mvLXCol[ xColPb ][ yColPb ], refIdxLXCol[ xColPb ][ yColPb ] and LX, respectively.
      * Otherwise, mvCol, refIdxCol, and listCol are set equal to mvLNCol[ xColPb ][ yColPb ], refIdxLNCol[ xColPb ][ yColPb ], and LN, respectively, with N being the value of collocated\_from\_l0\_flag.

and mvLXCol and availableFlagLXCol are derived as follows:

* + If LongTermRefPic( currPic, currPb, refIdxLX, LX ) is not equal to LongTermRefPic( colPic, colPb, refIdxCol, listCol ), both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
  + Otherwise, the variable availableFlagLXCol is set equal to 1, refPicListCol[ refIdxCol ] is set to be the picture with reference index refIdxCol in the reference picture list listCol of the slice containing prediction block currPb in the picture colPic, and the following applies:

colPocDiff = DiffPicOrderCnt( colPic, refPicListCol[ refIdxCol ] ) (8‑177)

currPocDiff = DiffPicOrderCnt( currPic, RefPicListX[ refIdxLX ] ) (8‑178)

* + - * If RefPicListX[ refIdxLX ] is a long-term reference picture, or colPocDiff is equal to currPocDiff, mvLXCol is derived as follows:

mvLXCol = mvCol (8‑179)

* + - * Otherwise, mvLXCol is derived as a scaled version of the motion vector mvCol as follows:

tx = ( 16384 + ( Abs( td )  >>  1 ) ) / td (8‑180)

distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 )  >>  6 ) (8‑181)

mvLXCol =  Clip3( −32768, 32767, Sign( distScaleFactor \* mvCol ) \*   
 ( ( Abs( distScaleFactor \* mvCol ) + 127 )  >>  8 ) ) (8‑182)

where td and tb are derived as follows:

td = Clip3( −128, 127, colPocDiff ) (8‑183)

tb = Clip3( −128, 127, currPocDiff ) (8‑184)

Method three:

This method directly changes the reference picture list selection of EL TMVP derivation process when ILP picture is enabled as the co-located picture for EL TMVP. This method has the minimal text changes to enable the proposed low-delay flag modification.

### 8.5.3.2.8 Derivation process for collocated motion vectors

Inputs to this process are:

* a variable currPb specifying the current prediction block,
* a variable colPic specifying the collocated picture,
* a variable colPb specifying the collocated prediction block inside the collocated picture specified by colPic,
* a luma location ( xColPb, yColPb ) specifying the top-left sample of the collocated luma prediction block specified by colPb relative to the top-left luma sample of the collocated picture specified by colPic,
* a reference index refIdxLX, with X being 0 or 1.

Outputs of this process are:

* the motion vector prediction mvLXCol,
* the availability flag availableFlagLXCol.

The variable currPic specifies the current picture.

The arrays predFlagLXCol[ x ][ y ], mvLXCol[ x ][ y ], and refIdxLXCol[ x ][ y ] are set equal to the corresponding arrays of the collocated picture specified by colPic, PredFlagLX[ x ][ y ], MvLX[ x ][ y ], and RefIdxLX[ x ][ y ], respectively, with X being the value of X this process is invoked for.

The variables mvLXCol and availableFlagLXCol are derived as follows:

* If colPb is coded in an intra prediction mode, both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
* Otherwise, the motion vector mvCol, the reference index refIdxCol, and the reference list identifier listCol are derived as follows:
  + If predFlagL0Col[ xColPb ][ yColPb ] is equal to 0, mvCol, refIdxCol, and listCol are set equal to mvL1Col[ xColPb ][ yColPb ], refIdxL1Col[ xColPb ][ yColPb ], and L1, respectively.
  + Otherwise, if predFlagL0Col[ xColPb ][ yColPb ] is equal to 1 and predFlagL1Col[ xColPb ][ yColPb ] is equal to 0, mvCol, refIdxCol, and listCol are set equal to mvL0Col[ xColPb ][ yColPb ], refIdxL0Col[ xColPb ][ yColPb ], and L0, respectively.
  + Otherwise (predFlagL0Col[ xColPb ][ yColPb ] is equal to 1 and predFlagL1Col[ xColPb ][ yColPb ] is equal to 1), the following assignments are made:
    - * If DiffPicOrderCnt( aPic, currPic ) is less than or equal to 0 for every picture aPic in every reference picture list of the current slice, or if DiffPicOrderCnt( colPic, currPic ) is equal to 0, mvCol, refIdxCol, and listCol are set equal to mvLXCol[ xColPb ][ yColPb ], refIdxLXCol[ xColPb ][ yColPb ] and LX, respectively.
      * Otherwise, mvCol, refIdxCol, and listCol are set equal to mvLNCol[ xColPb ][ yColPb ], refIdxLNCol[ xColPb ][ yColPb ], and LN, respectively, with N being the value of collocated\_from\_l0\_flag.

and mvLXCol and availableFlagLXCol are derived as follows:

* + If LongTermRefPic( currPic, currPb, refIdxLX, LX ) is not equal to LongTermRefPic( colPic, colPb, refIdxCol, listCol ), both components of mvLXCol are set equal to 0 and availableFlagLXCol is set equal to 0.
  + Otherwise, the variable availableFlagLXCol is set equal to 1, refPicListCol[ refIdxCol ] is set to be the picture with reference index refIdxCol in the reference picture list listCol of the slice containing prediction block currPb in the picture colPic, and the following applies:

colPocDiff = DiffPicOrderCnt( colPic, refPicListCol[ refIdxCol ] ) (8‑177)

currPocDiff = DiffPicOrderCnt( currPic, RefPicListX[ refIdxLX ] ) (8‑178)

* + - * If RefPicListX[ refIdxLX ] is a long-term reference picture, or colPocDiff is equal to currPocDiff, mvLXCol is derived as follows:

mvLXCol = mvCol (8‑179)

* + - * Otherwise, mvLXCol is derived as a scaled version of the motion vector mvCol as follows:

tx = ( 16384 + ( Abs( td )  >>  1 ) ) / td (8‑180)

distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 )  >>  6 ) (8‑181)

mvLXCol =  Clip3( −32768, 32767, Sign( distScaleFactor \* mvCol ) \*   
 ( ( Abs( distScaleFactor \* mvCol ) + 127 )  >>  8 ) ) (8‑182)

where td and tb are derived as follows:

td = Clip3( −128, 127, colPocDiff ) (8‑183)

tb = Clip3( −128, 127, currPocDiff ) (8‑184)