Restriction on motion vector scaling for Merge and AMVP (JCTVC-G551)

Il-Koo Kim, Youngo Park, Nikolay Shlyakhov and JeongHoon Park
(Samsung)

Introduction (1)

- H.264/AVC specifies **long-term reference pictures** for coding efficiency and error resilience.
  - Long-term reference pictures are **managed separately** with normal (short-term) reference pictures
    - Decoder cannot get correct picture order count (POC) of long term reference pictures, it is impossible to use the POCs of long-term reference pictures in motion vector scaling process.
    - When the **long-term references are involved** in getting motion vector for temporal direct mode, motion vector scaling is restricted, that is, motion vector of collocated block is not scaled to generate a motion vector of current block. Instead, the motion vector of collocated block is used **without scaling** as the motion vector of current block.

Introduction (2)

- Same concept of long-term reference pictures are inherited in HEVC
  - **But**, when long-term reference pictures are involved, motion vector scaling process is **not** clearly specified in the current working draft (WD) for both spatial and temporal motion vector prediction of Merge and AMVP.
- In AHG21 (Reference picture buffering and list construction)
  - One concern was raised that unless long-term reference pictures are managed separately with short-term reference pictures, there would be coding efficiency loss.
- To following the above discussion,
  - Coding efficiency impact is demonstrated when motion vector scaling is restricted under specific conditions **without distinguishing** between the motion vectors from long-term reference pictures and short-term reference pictures, respectively.

Scaling in H.264/AVC

❖ In temporal Direct mode

When reference index refers to long-term reference picture, **collocated MV is used as current MV without scaling**

– If the reference index refIdxL0 refers to a long-term reference picture, or DiffPicOrderCnt(pic1, pic0) is equal to 0, the motion vectors mvL0, mvL1 for the direct mode partition are derived by:

mvL0 = mvCol (8-197)

mvL1 = 0 (8-198)

– Otherwise, the motion vectors mvL0, mvL1 are derived as scaled versions of the motion vector mvCol of the co-located sub-macroblock partition as specified below (see Figure 8-2).

tx = (16 384 + Abs(td / 2)) / td (8-199)

DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6) (8-200)

mvL0 = (DistScaleFactor * mvCol + 128) >> 8 (8-201)

mvL1 = mvL0 − mvCol (8-202)

where tb and td are derived as:

tb = Clip3(-128, 127, DiffPicOrderCnt(currPicOrField, pic0)) (8-203)

td = Clip3(-128, 127, DiffPicOrderCnt(pic1, pic0)) (8-204)

NOTE 4 – mvL0 and mvL1 cannot exceed the ranges specified in Annex A.

The prediction utilization flags predFlagL0 and predFlagL1 are both set equal to 1.

Figure 8-2 illustrates the temporal direct-mode motion vector inference when the current picture is temporally between the reference picture from reference picture list 0 and the reference picture from reference picture list 1.

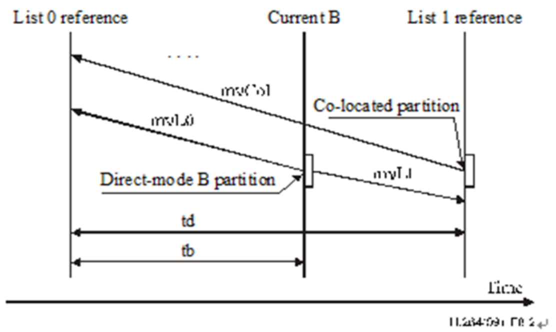
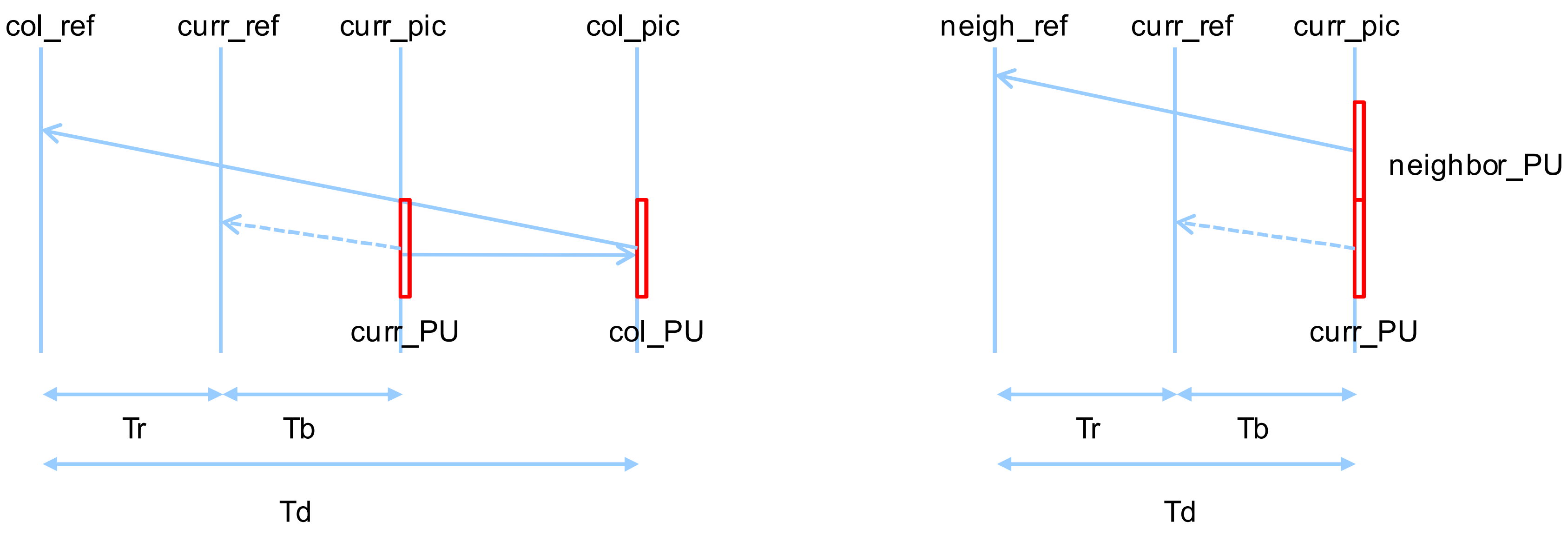


Figure 8-2 – Example for temporal direct-mode motion vector inference (informative)

MV scaling restrictions (1)

- Proposed method (1)
  - If POC difference between reference pictures of current picture and reference picture of candidate PU (**Tr**) is larger than threshold (**TH$_{poc_diff}$**), **the candidate PU is not used in scaling** for Merge and AMVP



(a) temporal scaling **(b) spatial scaling**

MV scaling restrictions (2)

- Proposed method (2)
  - **THpoc_diff** is adjusted according to the relative position of current picture and reference pictures.
    - **Case 1**) When both POCs of col_ref and curr_ref are larger than curr_pic or both POCs of col_ref and curr_ref are smaller than curr_pic
    - **Case 2**) When the POC of curr_pic is between POCs of col_ref and curr_ref.

```
#if MV_SCALING_RESTRICTION
    Int iDiffPocD = iNeibPOC - iNeibRefPOC;
    Int iDiffPocB = iCurrPOC - iCurrRefPOC;
    Int iTh;

    if ( (iDiffPocD < 0 && iDiffPocB < 0) || (iDiffPocD > 0 && iDiffPocB > 0) )
    {
      iTh = MV_SCALING_RESTRICTION_TH;
    }
    else
    {
      iTh = MV_SCALING_RESTRICTION_TH2;
    }
    if ( abs(iCurrRefPOC-iNeibRefPOC) > iTh )
    {
      return false;
    }
#endif

    Int iScale = xGetDistScaleFactor( iCurrPOC, iCurrRefPOC, iNeibPOC, iNeibRefPOC );
    if ( iScale == 1024 )
    {
      rcMv = cMvPred;
    }
    else
    {
      rcMv = cMvPred.scaleMv( iScale );

    }
```

Test Results (1)

- Test results under common test condition (JCTVC-F900)
  - **Anchor software: HM-4.0**
  - **Pre-defined thresholds**
    - **MV_SCALING_RESTRICTION_TH = 8**
    - **MV_SCALING_RESTRICTION_TH2 = 16**

Test Results (2)

- **Coding efficiency**
  - 0.0% (RAHE), 0.0% (RALC), 0.1% (LDHE), 0.0% (LDLC), 0.1% (LPHE), 0.1% (LPLC) with class F sequences.
  - Coding efficiency gain for Class F only is 0.13%.

	Random Access HE			Random Access LC		
	Y	U	V	Y	U	V
Class A	0.0%	0.0%	-0.1%	0.0%	0.1%	0.0%
Class B	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%
Class C	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class D	0.1%	0.1%	0.2%	0.1%	0.0%	0.0%
Class E						
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Overall	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	100%			101%		

	Low delay B HE			Low delay B LC		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	-0.1%	-0.2%	0.0%	0.0%	-0.2%
Class C	-0.1%	0.0%	0.0%	-0.1%	0.1%	-0.3%
Class D	-0.1%	0.0%	0.1%	0.0%	0.0%	-0.1%
Class E	0.0%	-0.2%	0.5%	-0.1%	0.1%	-0.4%
Class F	-0.3%	-0.2%	0.1%	0.0%	-0.1%	-0.2%
Overall	-0.1%	-0.1%	0.1%	0.0%	0.0%	-0.2%
	-0.1%	-0.1%	0.0%	0.0%	0.0%	-0.2%
Enc Time[%]	100%			100%		
Dec Time[%]	101%			101%		

	Low delay P HE			Low delay P LC		
	Y	U	V	Y	U	V
Class A						
Class B	-0.1%	-0.1%	0.2%	0.0%	0.0%	-0.2%
Class C	-0.1%	-0.1%	-0.2%	0.0%	-0.1%	0.0%
Class D	0.0%	0.1%	-0.4%	-0.1%	-0.1%	-0.2%
Class E	0.0%	-0.4%	0.1%	0.0%	-0.5%	-0.3%
Class F	-0.2%	0.0%	-0.4%	-0.1%	-0.1%	0.3%
Overall	-0.1%	-0.1%	-0.1%	-0.1%	-0.1%	-0.1%
	-0.1%	-0.1%	-0.1%	-0.1%	-0.1%	-0.1%
Enc Time[%]	100%			100%		
Dec Time[%]	101%			100%		

cross-checked by HiSilicon (JCTVC-G859)

Thanks.

Conclusions

- A restriction method on motion vector scaling was proposed.
  - When POC difference (Tr) between reference picture of current PU and reference picture of candidate PU (collocated PU or neighbor PU) are larger than pre-determined threshold (*THpoc_diff*), **motion vector predictor is marked as unavailable** before the scaling process is performed.
  - This restriction is applied to both **spatial and temporal scaling**.
- Average coding efficiency gain
  - 0.0% (RAHE), 0.0% (RALC), 0.1% (LDHE), 0.0% (LDLC), 0.1% (LPHE), 0.1% (LPLC) with class F sequences.
  - Coding efficiency gain for Class F only is 0.13%.
- The results demonstrate that efficient scaling is possible **without** long-term and short-term reference pictures **separation**.
- It is recommended to adopt this restriction process in HEVC.