

Restricted usage of motion vectors for long-term reference picture in motion vector prediction process

(JCTVC-J0302)

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

MV Scaling in current HEVC

❖ HM-7.1

- Spatial and temporal MV scaling
 - if(current PU uses LTRP || neighboring(or co-located) PU uses LTRP) {
 - ⊕ **Use non-scaled MV;**
 - }
 - else {
 - ⊕ Use scaled MV;
 - }
 - return true; // mark as **available**
- There is no way to mark as **unavailable** when scaling is involved

Proposed MV Scaling

❖ JCTVC-J0302 → There are ways to mark as **unavailable**, and **non-scaled MV is not used**.

- Solution 1 for spatial and temporal MV scaling

- if(current PU uses LTRP || neighboring(or co-located PU) uses LTRP) {
 - ⊕ return false; // mark as **unavailable**
- }
- else {
 - ⊕ Use scaled MV;
 - ⊕ return true; // mark as **available**
- }

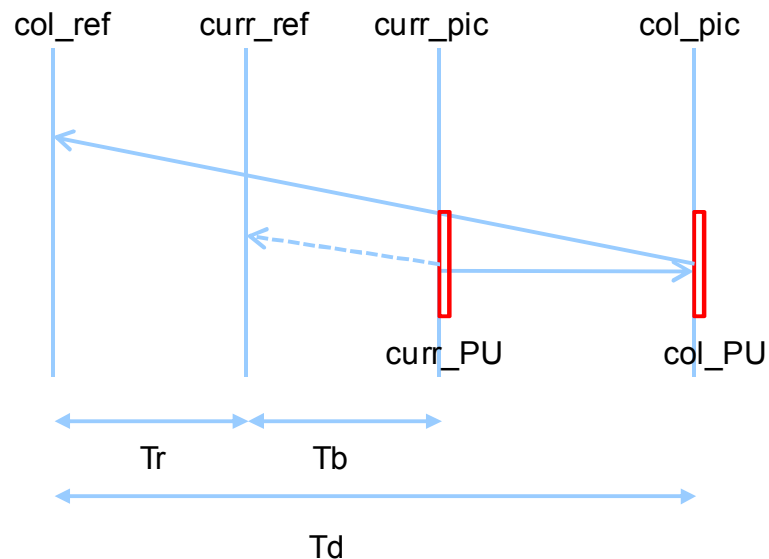
- Solution 2 for spatial and temporal MV scaling

- if($Tr \geq \text{threshold}$) {
 - ⊕ return false; // mark as **unavailable**
- }
- else {
 - ⊕ Use scaled MV;
 - ⊕ return true; // mark as **available**
- }

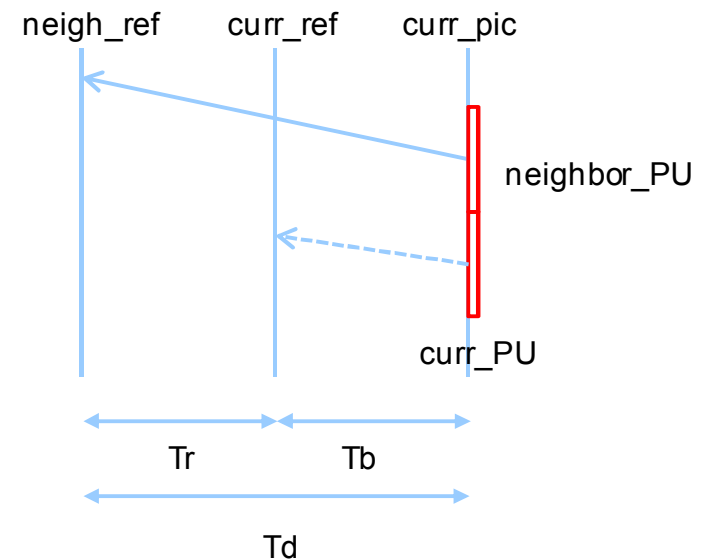
MV scaling restrictions (1)

❖ Proposed solution 2 (1)

- If POC difference between reference pictures of current picture and reference picture of candidate PU (T_r) is larger than threshold (TH_{poc_diff}), **motion vector predictor is marked as unavailable** before the scaling process is performed.



(a) temporal scaling



(b) spatial scaling

MV scaling restrictions (2)

❖ Proposed solution 2 (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 );
    }
}
```

MV scaling restrictions (3)

❖ Benefits of proposed solutions

- The benefit of these approach is that by removing the inefficient motion vectors which are scaled or non-scaled motion vectors from LTRPs, more efficient motion vectors can be included in the list.
- These methods can be utilized for temporal motion vector predictor for Merge and AMVP process, and spatial motion vector prediction for AMVP process.

Test Results (1)

- ❖ Test results under common test condition (JCTVC-I1100)
 - **Anchor software: HM-7.1**
 - **Pre-defined thresholds**
 - MV_SCALING_RESTRICTION_TH = 8
 - MV_SCALING_RESTRICTION_TH2 = 16

Test Results (2)

❖ Coding efficiency

- Slight coding efficiency impact under common test condition
 - Coding efficiency gain for Class F low delay conditions is about 0.1%.
- Runtime measurement is inaccurate.

	Random Access Main			Random Access HE10		
	Y	U	V	Y	U	V
Class A	0.0%	0.0%	-0.1%	0.0%	-0.2%	0.1%
Class B	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Class C	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%
Class D	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%
Class E						
Overall	0.0%	0.0%	0.0%	0.0%	-0.1%	0.0%
	0.0%	0.0%	0.0%	0.0%	-0.1%	0.0%
Class F	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Enc Time[%]	100%			100%		
Dec Time[%]	102%			99%		

	Low delay B Main			Low delay B HE10		
	Y	U	V	Y	U	V
Class A						
Class B	0.0%	0.1%	-0.1%	0.0%	0.0%	0.1%
Class C	0.0%	0.0%	-0.1%	-0.1%	-0.2%	-0.1%
Class D	-0.1%	-0.2%	0.3%	0.0%	-0.1%	-0.3%
Class E	-0.1%	-0.6%	-0.3%	0.0%	-0.1%	0.0%
Overall	0.0%	-0.1%	0.0%	0.0%	-0.1%	-0.1%
	0.0%	-0.1%	0.0%	0.0%	0.0%	0.0%
Class F	-0.1%	-0.1%	0.1%	-0.1%	0.0%	1.0%
Enc Time[%]	101%			101%		
Dec Time[%]	100%			98%		

**Cross-checked by Sony
(JCTVC-J0341)**

Thanks.

Conclusions

- ❖ Two solutions on motion vector scaling were proposed when LTRPs are involved.
- ❖ Solution 1
 - The first solution is not to insert motion vectors into the candidate list **by marking it unavailable** if the motion vectors are from LTRPs.
- ❖ Solution 2
 - 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 (TH_{poc_diff}), **motion vector predictor is marked as unavailable** before the scaling process is performed.
- ❖ Average coding efficiency gain of solution 2
 - Slight coding efficiency impact under common test condition
 - Coding efficiency gain for Class F low delay conditions is about 0.1%.
- ❖ It is recommended to adopt one of these restrictions in HEVC.

MV Scaling in JCTVC-J0071

❖ JCTVC-J0071

■ Spatial MV scaling

- if(current and neighboring PUs use **same** type of reference picture) {
 - ⊕ if(current PU uses LTRP || neighboring PU uses LTRP) {
 - ◆ **Use non-scaled MV;**
 - ⊕ }
 - ⊕ else {
 - ◆ Use scaled MV;
 - ⊕ }
 - ⊕ return true; // mark as available
- }
- else {
 - ⊕ return false; // mark as unavailable
- }

- There are ways to mark as **unavailable**, but still **non-scaled MV is used**.

MV Scaling in JCTVC-J0071

❖ JCTVC-J0071

■ Temporal MV scaling

- if(current and co-located PUs use **different** type of reference picture) {
 - ⊕ return false; // mark as unavailable
- }
- else {
 - ⊕ if(current PU uses LTRP || co-located PU uses LTRP) {
 - ◆ **Use non-scaled MV;**
 - ⊕ }
 - ⊕ else {
 - ◆ Use scaled MV;
 - ⊕ }
 - ⊕ return true; // mark as **available**
- }

- There are ways to mark as **unavailable**, but still **non-scaled MV is used**.