JCTVC-I0109

New results of implicit weighted prediction

[information]

Akiyuki Tanizawa
Takeshi Chujoh
Tomoo Yamakage

TOSHIBA
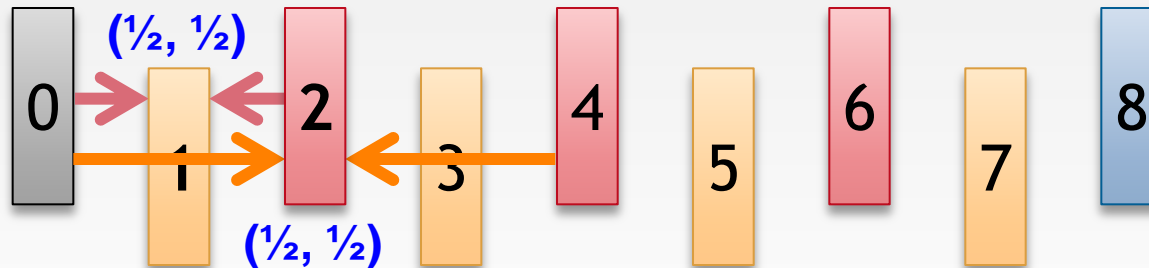**Leading Innovation** >>>

Summary

- AHG18: Weighted Prediction in HEVC
  - AVC based WP (AVCWP) was adopted in HM4/WD4

- Weighted prediction of HM6.0
  - The gain of implicit WP is not so high compared with the explicit WP, especially RA and LB cases.

- IBBP coding structure
  - AVC-based implicit WP is originally managed for IBBP coding structure, However current HM6.0 does not support such kinds of structures.

- Results:
  - Report informative experimental results on IBBBP (M=4) coding structures
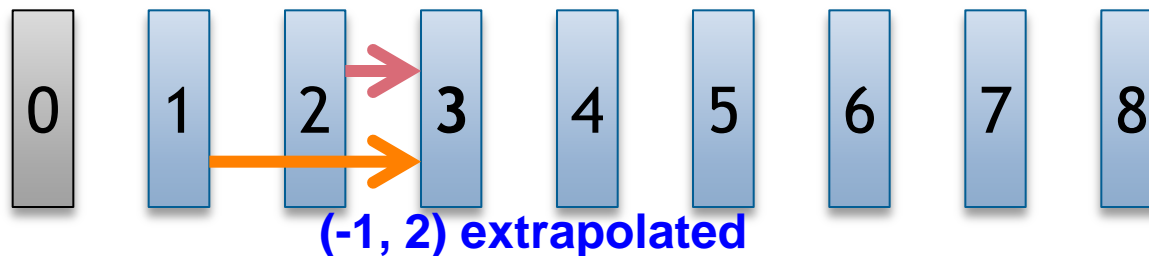
Implicit WP on several coding structures

- Random Access (RA) case
  - Weighting factors of closest references are ½ (same as default)

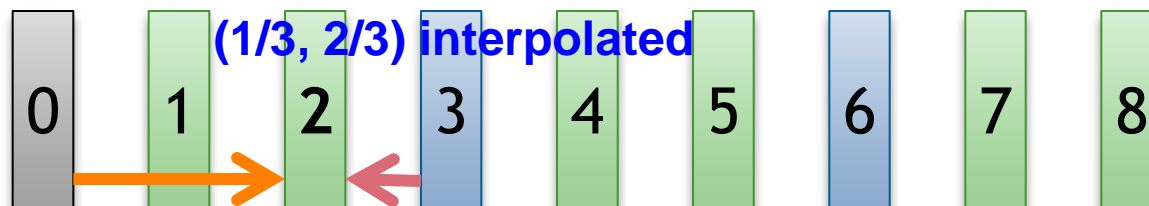(½, ½)

(½, ½)

- Low Delay (LD) case
  - Forward B-slices only, Weighting factors can be extrapolated

(-1, 2) extrapolated

- IBBP (M=3) case
  - Weighting factors are (1/3, 2/3) or (2/3, 1/3)

(1/3, 2/3) interpolated

Motion estimation on implicit WP

- Motion estimation in HM6.0
  - Uni-pred: uni-pred cost is calculated
  - Bi-pred: two uni-pred costs for list l0 and l1 are calculated repeatedly
- Cost function in HM6.0
  - Default ME cost

    $SAD \mathrel{+}= abs(Org[n] - Pred[n]);$

  - Weighted ME cost

    $Pred_{WP} = ((w_0*Pred[n] + round) >> shift) + o_0;$

    $SAD \mathrel{+}= abs(Org[n] - Pred_{WP});$

When implicit WP enables, w_0 is always 1 and o_0 is always 0. However, Weighted ME cost is used. This may cause encoding time increase!

TOSHIBA
Leading Innovation >>>

Experimental results on IBBBP (M=4) structure

Black-fade sequences

	Implicit WP **with** Weighted ME						Implicit WP **without** Weighted ME					
	Random Access Main			**Random Access HE10**			**Random Access Main**			**Random Access HE10**		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class A	-2.6%	-1.7%	-2.0%	-2.9%	-2.2%	-2.3%	-2.6%	-1.7%	-2.0%	-2.9%	-2.2%	-2.3%
Class B	-5.2%	-4.8%	-4.7%	-5.2%	-5.6%	-5.0%	-5.2%	-4.8%	-4.7%	-5.2%	-5.6%	-4.9%
Class C	-1.8%	-1.8%	-2.2%	-1.9%	-2.6%	-2.7%	-1.8%	-1.7%	-2.2%	-1.9%	-2.4%	-2.6%
Class D	-3.1%	-2.5%	-3.3%	-3.1%	-3.2%	-3.6%	-3.1%	-2.5%	-3.2%	-3.1%	-3.2%	-3.6%
Class E												
Overall	-3.31%	-2.8%	-3.2%	-3.38%	-3.5%	-3.5%	-3.28%	-2.8%	-3.1%	-3.37%	-3.5%	-3.5%
	-3.3%	-2.8%	-3.2%	-3.4%	-3.5%	-3.5%	-3.3%	-2.8%	-3.1%	-3.4%	-3.5%	-3.5%
Enc Time[%]	130%			123%			94%			92%		
Dec Time[%]	92%			93%			92%			93%		

White-fade sequences

	Implicit WP **with** Weighted ME						Implicit WP **without** Weighted ME					
	Random Access Main			**Random Access HE10**			**Random Access Main**			**Random Access HE10**		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class A	-3.5%	-2.8%	-3.1%	-3.7%	-3.2%	-3.4%	-3.5%	-2.8%	-3.1%	-3.7%	-3.2%	-3.4%
Class B	-5.5%	-5.6%	-5.7%	-5.5%	-6.7%	-6.0%	-5.5%	-5.6%	-5.7%	-5.4%	-6.7%	-5.9%
Class C	-1.9%	-2.0%	-2.3%	-1.9%	-2.6%	-2.6%	-1.9%	-1.9%	-2.3%	-1.9%	-2.6%	-2.6%
Class D	-3.0%	-2.5%	-3.2%	-2.9%	-3.2%	-3.5%	-2.9%	-2.4%	-3.2%	-2.9%	-3.2%	-3.5%
Class E												
Overall	-3.57%	-3.3%	-3.7%	-3.62%	-4.1%	-4.0%	-3.55%	-3.3%	-3.7%	-3.60%	-4.1%	-4.0%
	-3.6%	-3.4%	-3.8%	-3.6%	-4.1%	-4.0%	-3.6%	-3.3%	-3.7%	-3.6%	-4.1%	-4.0%
Enc Time[%]	134%			126%			94%			91%		
Dec Time[%]	91%			92%			90%			92%		

Implicit WP is effective for IBBBP (M=4) coding structure (and maybe M=3, too) and Weighted ME for implicit WP is not needed.

Conclusion

- Experimental Results on IBBBP coding structure:
  - For Black-fade and White-fade sequences
    - M4-Main: 3.3%/3.6%
    - M4-HE: 3.4%/3.6%
    - This scheme can reduce the decoding runtime by 7 to 10%
    - It is not necessary to apply to the weighted ME for implicit WP

- Suggestion;
  - This 1 line modification is applied to the next version of HM.

```
Function;
Void  TEncSearch::setWpScalingDistParam( TComDataCU* pcCU, Int iRefIdx, RefPicList
    eRefPicListCur )
 -----------Original-------------------
m_cDistParam.bApplyWeight = ( pcSlice->getSliceType()==P_SLICE && pps->getUseWP() ) ||
    ( pcSlice->getSliceType()==B_SLICE && pps->getWPBiPredIdc() ) ;
 ------------Modified-----------------
m_cDistParam.bApplyWeight = ( pcSlice->getSliceType()==P_SLICE && pps->getUseWP() ) ||
    ( pcSlice->getSliceType()==B_SLICE && (pps->getWPBiPredIdc() == 1) ) ;
```