

JCTVC-J0055

AHG9: WEIGHTED PREDICTION PARAMETER SIGNALLING

Y. He, Y. Ye

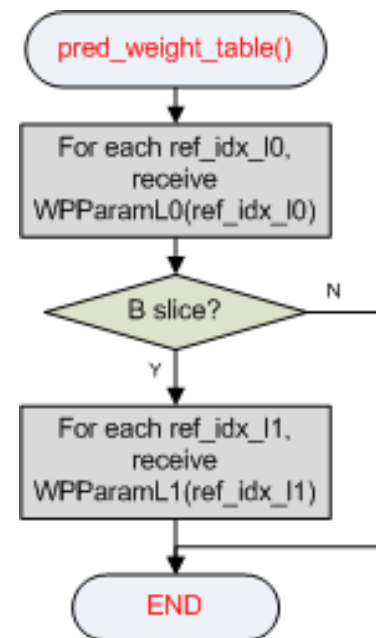
InterDigital Communications, LLC

10th JCT-VC meeting, July 10-20, 2012

Weighted Prediction Parameter Signalling in WD7

- At the 9th meeting, list combination was removed (JCTVC-I0125)
- WP parameter signalling restored to the same as in H.264/AVC
- Signalling redundancy can be significant
 - Duplicate L0/L1 entries often share the same WP parameters
 - In LDB, all pictures have identical entries on L0 and L1
 - In RA, ~50% L0 and L1 entries are duplicate of each other

POC	RPS		Ref pic lists	
	StCurr0	StCurr1	L0	L1
16	{8, 6, 4, 0}	{NULL}	{8, 6, 4, 0}	{8, 6, 4, 0}
12	{8, 6}	{16}	{8, 6}	{16, 8}
10	{8, 6}	{12, 16}	{8, 6}	{12, 16}
9	{8}	{10, 12, 16}	{8, 10}	{10, 12}
11	{10, 8}	{12, 16}	{10, 8}	{12, 16}
14	{12, 10, 8}	{16}	{12, 10}	{16, 12}
13	{12, 8}	{14, 16}	{12, 8}	{14, 16}
15	{14, 12, 8}	{16}	{14, 12}	{16, 14}



Proposed Method 1

- Method 1: add new syntax element **weights_l1_present_flag**
 - If **weights_l1_present_flag** = 0, entire L1 WP parameter signalling is skipped
 - Conditioned upon L0 and L1 size being equal: no need to construct L0 and L1 before parsing **pred_weight_table()**

pred_weight_table() {	
....	
if(slice_type == B) {	
if (num_ref_idx_l1_active_minus1 == num_ref_idx_l0_active_minus1)	
weights_l1_present_flag	u(1)
if (weights_l1_present_flag)	
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {	
luma_weight_l1_flag	u(1)
if(luma_weight_l1_flag) {	
delta_luma_weight_l1[i]	se(v)
luma_offset_l1[i]	se(v)
}	
if(chroma_format_idc != 0) {	
chroma_weight_l1_flag	u(1)
if(chroma_weight_l1_flag)	
for(j = 0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	
}	
}	
...	

weights_l1_present_flag equal to 1 specifies that syntax elements used to derive the weighting factors **LumaWeightL1[i]** and **ChromaWeightL1[i][j]** and the additive offsets **LumaOffsetL1[i]** and **ChromaOffsetL1[i][j]** for the luma and chroma prediction values of list 1 prediction are present. When **weights_l1_present_flag** is equal to 0, syntax elements used to derive the weighting factors and the additive offsets for luma and chroma prediction values of list 1 prediction are not present. Instead, the weighting factors **LumaWeightL1[i]** and **ChromaWeightL1[i][j]** and the additive offsets **LumaOffsetL1[i]** and **ChromaOffsetL1[i][j]** are inferred by invoking the decoding process in 8.3.6. When **weights_l1_present_flag** is not present, it is inferred to be 1.

Decoding Process of Method 1

1. Set blIdenticalListFlag according to the following:

- If PicOrderCnt(RefPicList0 [i]) and PicOrderCnt(RefPicList1 [i]) are the same for all entries in list 0 and list 1, then blIdenticalListFlag is set to equal to 1
- Otherwise, blIdenticalListFlag is set to equal to 0

2. Use the following pseudo code to derive L1 weights and offsets:

```
for( i = 0; i ≤ num_ref_idx_l1_active_minus1; i++ ) {  
    if ( blIdenticalListFlag ) {  
        LumaWeightL1Flag[ i ] = LumaWeightL0Flag[ i ]  
        LumaWeightL1[ i ] = LumaWeightL0[ i ]  
        LumaOffsetL1[ i ] = LumaOffsetL0[ i ]  
        ChromaWeightL1Flag[ i ] = ChromaWeightL0Flag[ i ]  
        ChromaWeightL1[ i ][ 0 ] = ChromaWeightL0[ i ][ 0 ]  
        ChromaWeightL1[ i ][ 1 ] = ChromaWeightL0[ i ][ 1 ]  
        ChromaOffsetL1[ i ][ 0 ] = ChromaOffsetL0[ i ][ 0 ]  
        ChromaOffsetL1[ i ][ 1 ] = ChromaOffsetL0[ i ][ 1 ]  
    }  
    else {  
        LumaWeightL1Flag[ i ] = 0  
        LumaWeightL1[ i ] = ( 1 << luma_log2_weight_denom )  
        LumaOffsetL1[ i ] = 0  
        ChromaWeightL1Flag[ i ] = 0  
        ChromaWeightL1[ i ][ 0 ] = ( 1 << ChromaLog2WeightDenom )  
        ChromaWeightL1[ i ][ 1 ] = ( 1 << ChromaLog2WeightDenom )  
        ChromaOffsetL1[ i ][ 0 ] = 0  
        ChromaOffsetL1[ i ][ 1 ] = 0  
    }  
}
```

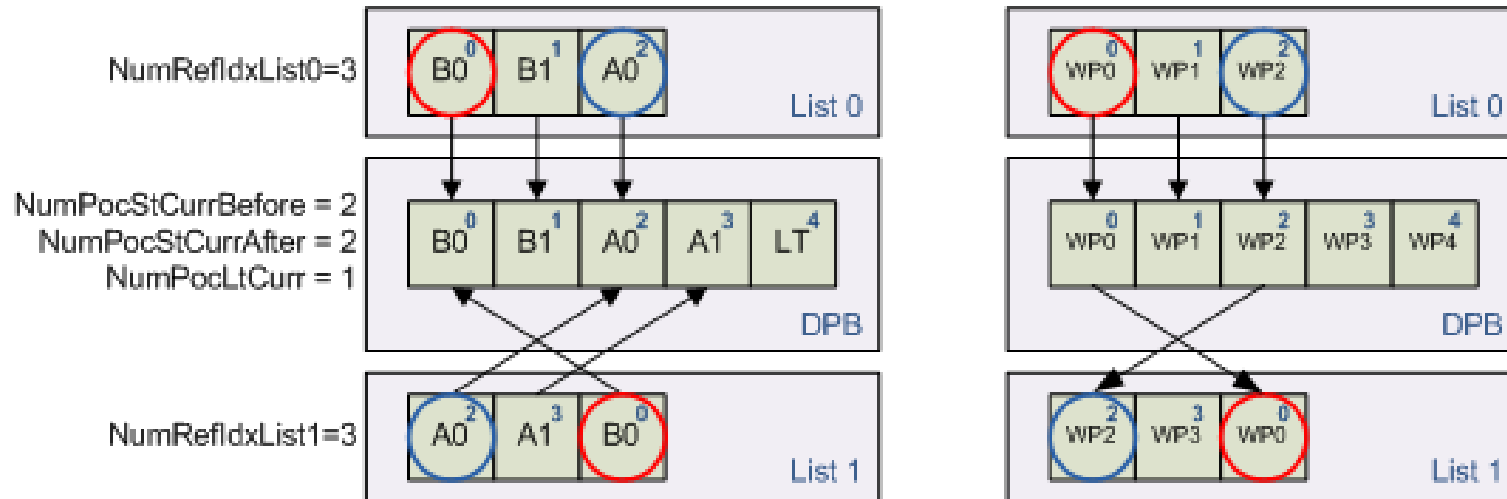
Proposed Method 2

- Method 2: method 1 + additional syntax **delta_params_present_flag[i]** for each L1 entry
 - Skip WP param signalling for the duplicate L1 entries on entry-by-entry basis
 - If **delta_params_present_flag[i] = 0**, WP parameters for i-th L1 entry is inferred from previously signalled WP parameters of the same ref pic

pred_weight_table () {	
....	
if(slice_type == B) {	
if(num_ref_idx_l1_active_minus1 == num_ref_idx_l0_active_minus1)	
weights_l1_present_flag	u(1)
if(weights_l1_present_flag)	
for(i=0; i <= num_ref_idx_l1_active_minus1; i++) {	
delta_params_present_flag[i]	u(1)
if(delta_params_present_flag[i]) {	
luma_weight_l1_flag[i]	u(1)
if(luma_weight_l1_flag[i]) {	
delta_luma_weight_l1[i]	se(v)
delta_luma_offset_l1[i]	se(v)
}	
if(chroma_format_idc != 0) {	
chroma_weight_l1_flag[i]	u(1)
if(chroma_weight_l1_flag[i])	
for(j=0; j < 2; j++) {	
delta_chroma_weight_l1[i][j]	se(v)
delta_chroma_offset_l1[i][j]	se(v)
}	
}	
}	
}	
}	
....	

delta_params_present_flag[i] equal to 1 specifies that the syntax elements **luma_weight_l1_flag**, **delta_luma_weight_l1**, **delta_luma_offset_l1**, **chroma_weight_l1_flag**, **delta_chroma_weight_l1**, **delta_chroma_offset_l1** are present. When **delta_params_present_flag[i]** is equal to 0, the syntax elements **luma_weight_l1_flag**, **delta_luma_weight_l1**, **delta_luma_offset_l1**, **chroma_weight_l1_flag**, **delta_chroma_weight_l1**, **delta_chroma_offset_l1** are not present.

Mapping of L0 and L1 entries and WP parameters



- WP parameters are stored for each DPB reference used to predict the current slice
 - Initialized to default values: $(w, o) = (1 \ll \log_2 WD, 0)$
- Updated as actual WP parameters for a given picture are received
- Less signalling overhead since mapping is inferred
- Automatically allows prediction of WP parameter values
- Longer decoding process

Simulation Results (fade-to-white)

	Method 1				Method 2	
	RA main	RA HE10	LDB main	LDB HE10	RA main	RA HE10
Class A	0.0%	0.0%			0.0%	0.0%
Class B	0.0%	-0.1%	-0.4%	-0.6%	-0.1%	-0.1%
Class C	-0.1%	-0.1%	-0.9%	-1.3%	-0.2%	-0.3%
Class D	-0.3%	-0.5%	-2.8%	-3.9%	-0.6%	-0.9%
Class E			-3.0%	-4.5%		
Overall	-0.1%	-0.2%	-1.6%	-2.3%	-0.2%	-0.3%

* Method 2 LBD results not included, as they are bit exact with method 1 LBD results

Simulation Results (fade-to-black)

	Method 1				Method 2	
	RA main	RA HE10	LDB main	LDB HE10	RA main	RA HE10
Class A	0.0%	0.0%			0.0%	0.0%
Class B	0.0%	0.0%	-0.3%	-0.5%	-0.1%	-0.1%
Class C	-0.1%	-0.1%	-0.7%	-1.1%	-0.1%	-0.2%
Class D	-0.2%	-0.4%	-2.2%	-3.4%	-0.4%	-0.7%
Class E			-2.3%	-3.9%		
Overall	-0.1%	-0.2%	-1.3%	-2.0%	-0.2%	-0.3%

We thank Toshiba for cross checking our results (JCTVC-J0223)

Conclusions

- Two methods to reduce WP parameter signalling redundancy in WD7
 - Method 1: allows to skip WP parameter signalling for the entire L1
 - Method 2: allows to skip WP parameter signalling for L1 on entry-by-entry basis
- 1.6% and 2.3% average luma BD rate reduction for “LDB main” and “LDB HE10”
- 0.2% and 0.3% average luma BD rate reduction for “RA main” and “RA HE10”
 - Including additional 0.1% average BD rate reduction using method 2
- WD text changes for both methods provided based on I1003_d5
- Recommend to adopt