Feb.1-10, 2012

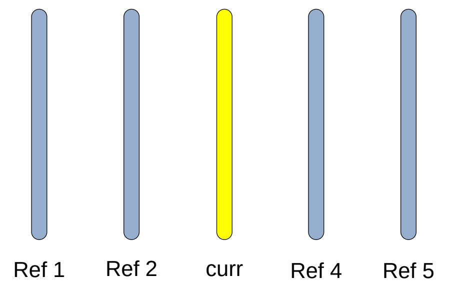
JCTVC-H0138 AHG21:Unification of reference picture list modification processes

Yong He, Yan Ye
InterDigital Communications, LLC

Introduction

- Unification of reference picture list modification process
  - Two different methods are currently used for L0/L1 modification and for LC modification
  - LC modification is more straightforward
  - Propose to use similar methods on L0/L1 modification
  - Simplifies syntax, semantics, and decoding process for L0/L1 modification
- Implemented an L0/L1 reordering method in HM5.0 encoder
  - Maximizes unique entries in L0 and L1
- Added bit counting for ref_pic_list_modification() in HM5.0 according to JCTVC-G1036
  - 47% bit reduction using proposed method for ref_pic_list_modification()

Reference picture list modification syntax (WD5.0)



RefPicSetStCurr0	Ref 2	Ref 1		
RefPicSetStCurr1	Ref 4	Ref 5		
Default List 0	Ref 2	Ref 1	Ref 4	Ref 5
Target modified List 0	Ref 2	Ref 1	Ref 4	Ref 2

```
ref_pic_list_modification( ) {                                   Descriptor
  if( slice_type != 2 ) {  // P slice or B slice
    ref_pic_list_modification_flag_l0                            u(1)
    if( ref_pic_list_modification_flag_l0 )
      do {
        list_modification_idc                                    ue(v)
        if( list_modification_idc != 3 )
          ref_pic_set_idx                                        ue(v)
      } while( list_modification_idc != 3 )
  }
  if( slice_type == 1 ) { // B slice
    ref_pic_list_modification_flag_l1                            u(1)
    if( ref_pic_list_modification_flag_l1 )
      do {
        list_modification_idc                                    ue(v)
        if( list_modification_idc != 3 )
          ref_pic_set_idx                                        ue(v)
      } while( list_modification_idc != 3 )
  }
}
```

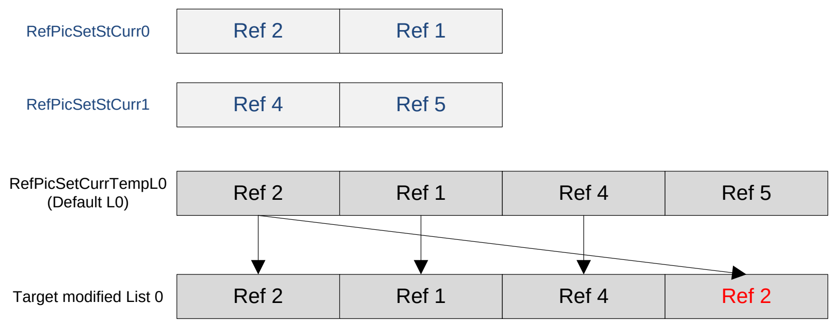
Step 1	Ref 2	Ref 1	Ref 4	Ref 5
Step 2	Ref 2	Ref 1	Ref 4	Ref 5
Step 3	Ref 2	Ref 1	Ref 4	Ref 5
Step 4	Ref 2	Ref 1	Ref 4	Ref 2
Step 5		finish		

ref_pic_list_modification syntax

RefIdx	list_modification_idc	ref_pic_set_idx
0	0	0
1	0	1
2	1	0
3	0	0
	3	

Proposed reference picture list modification process

- In comparison, LC modification does not use list_modification_idx, instead each entry in the modified LC is signaled explicitly
- Propose to unify the L0/L1 modification process with LC modification
- Remove list_modification_idc, and use ref_pic_set_idx to indicate the entry into temporary arrays RefPicSetCurrTempList0/1



Proposed ref_pic_list_modification syntax

RefIdx	ref_pic_set_idx
0	0
1	1
2	2
3	0

ref_pic_list_modification( ) {	**Descriptor**
if(slice_type != 2) { // P slice or B slice	
ref_pic_list_modification_flag_l0	u(1)
if(ref_pic_list_modification_flag_l0)	
~~do {~~	
~~**ref_pic_list_modification_idc**~~	~~ue(v)~~
~~if(ref_pic_list_modification_idc != 3)~~	
~~**ref_pic_set_idx**~~	~~ue(v)~~
~~} while(ref_pic_list_modification_idc != 3)~~	
for (i =0; i <= num_ref_idx_l0_active_minus1; i++) {	
if (NumRpsCurrTempList0 > 1)	
ref_pic_set_idx	te(v)
}	
}	
if(slice_type = = 1) { // B slice	
ref_pic_list_modification_flag_l1	u(1)
if(ref_pic_list_modification_flag_l1)	
~~do {~~	
~~**ref_pic_list_modification_idc**~~	~~ue(v)~~
~~if(ref_pic_list_modification_idc != 3)~~	
~~**ref_pic_set_idx**~~	~~ue(v)~~
~~} while(ref_pic_list_modification_idc != 3)~~	
for (i =0; i <= num_ref_idx_l1_active_minus1; i++) {	
if (NumRpsCurrTempList1 > 1)	
ref_pic_set_idx	te(v)
}	
}	
}	

Proposed semantics

ref_pic_list_modification_flag_l0 equal to 1 specifies that the syntax element ~~ref_pic_list_modification_idc~~ ref_pic_set_idx is present for specifying reference picture list 0. ref_pic_list_modification_flag_l0 equal to 0 specifies that this syntax element is not present.

~~When ref_pic_list_modification_flag_l0 is equal to 1, the number of times that ref_pic_list_modification_idc is not equal to 3 following ref_pic_list_modification_flag_l0 shall not exceed num_ref_idx_l0_active_minus1 + 1.~~

ref_pic_list_modification_flag_l1 equal to 1 specifies that the syntax element ~~ref_pic_list_modification_idc~~ ref_pic_set_idx is present for specifying reference picture list 1. ref_pic_list_modification_flag_l1 equal to 0 specifies that this syntax element is not present.

~~When ref_pic_list_modification_flag_l1 is equal to 1, the number of times that ref_pic_list_modification_idc is not equal to 3 following ref_pic_list_modification_flag_l1 shall not exceed num_ref_idx_l1_active_minus1 + 1.~~

~~**ref_pic_list_modification_idc** together with ref_pic_set_idx specifies which of the reference pictures are re-mapped. The values of ref_pic_list_modification_idc are specified in Table 7-4. The value of the first ref_pic_list_modification_idc that follows immediately after ref_pic_list_modification_flag_l0 or ref_pic_list_modification_flag_l1 shall not be equal to 3.~~

~~**Table 7-6 – ref_pic_list_modification_idc operations for modification of reference picture lists**~~

~~**ref_pic_list_modification_idc**~~	~~**modification specified**~~
~~0~~	~~For list 0: ref_pic_set_idx is present and corresponds to an index to RefPicSetStCurr0; For list 1: ref_pic_set_idx is present and corresponds to an index to RefPicSetStCurr1~~
~~1~~	~~For list 0: ref_pic_set_idx is present and corresponds to an index to RefPicSetStCurr1; For list 1: ref_pic_set_idx is present and corresponds to an index to RefPicSetStCurr0~~
~~2~~	~~ref_pic_set_idx is present and corresponds to an index to RefPicSetLtCurr~~
~~3~~	~~End loop for modification of the initial reference picture list~~

ref_pic_set_idx specifies the index~~, to RefPicSetStCurr0, RefPicSetStCurr1 or RefPicSetLtCurr,~~ of the reference picture ~~being moved to the current index in the reference picture list~~ in RefPicSetCurrTempListX to be placed at the current position of reference picture list LX. The value of ref_pic_set_idx shall be in the range of 0 to max_num_ref_frames, inclusive. If the syntax element ref_pic_set_idx is not present, it is set to 0.

Proposed decoding process for list initialization

1. RefPicSetCurrTempList0 is constructed from RefPicSetStCurr0, RefPicSetStCurr1 and RefPicSetLtCurr.

```
cIdx = 0
NumRpsCurrTempList0 = NumRpsStCurr0 + NumRpsStCurr1 + NumRpsLtCurr
if (NumRpsCurrTempList0 <= num_ref_idx_l0_active_minus1)
   NumRpsCurrTempList0 = num_ref_idx_l0_active_minus1+1
while( cIdx < NumRpsCurrTempList0 )
{
   for( i=0; i < NumPocStCurr0 && cIdx < NumRpsCurrTempList0; cIdx++, i++ )
    RefPicSetCurrTempList0 [ cIdx ] = RefPicSetStCurr0[ i ]
   for( i=0;  i < NumPocStCurr1 && cIdx < NumRpsCurrTempList0; cIdx++, i++ )
    RefPicSetCurrTempList0 [ cIdx ] = RefPicSetStCurr1[ i ]
   for( i=0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0; cIdx++, i++ )
    RefPicSetCurrTempList0 [ cIdx ] = RefPicSetLtCurr[ i ]
}
```

2. If ref_pic_list_modification_flag_l0 is 0, the initial RefPicList0 is constructed by taking the first num_ref_idx_l0_active_minus1+1 entries from RefPicSetCurrTempList0.

RefPicSetCurr0	Ref2	Ref1		
RefPicSetCurr1	Ref4	Ref5		
RefPicSetCurrTempL0 (default L0)	Ref2	Ref1	Ref4	Ref5
RefPicSetCurrTempL1 (default L1)	Ref4	Ref5	Ref2	Ref1

Proposed decoding process for list modification

Proposed modification process for reference picture lists

Input to this process is an array of reference picture RefPicSetCurrTempLX, and the size of the reference picture list num_ref_idx_lX_active_minus1 (with X being 0 or 1).

Output of this process is an array containing the modified reference picture list RefPicListX.

Let refIdxLX be an index into the reference picture list RefPicListLX. It is initially set equal to 0.

The following process is repeated until refIdxLX is greater than num_ref_idx_lX_active_minus1+1.

– RefPicListX [refIdxLX++] = RefPicSetCurrTempLX [ref_pic_set_idx]

~~Modification process for reference picture lists~~

~~After the invocation of this process, there shall be no reference pictures with greater temporal_id than the current slice included in the output RefPicList0 or RefPicList1.~~

~~When ref_pic_list_modification_flag_l0 is equal to 1, the following applies:~~

~~Let refIdxL0 be an index into the reference picture list RefPicList0. It is initially set equal to 0.~~

~~The corresponding syntax elements modification_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.~~

- ~~If modification_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause 8.2.2.3.1 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.~~
- ~~Otherwise, if modification_of_pic_nums_idc is equal to 2, the process specified in subclause 8.2.2.3.2 is invoked with refIdxL0 as input, and the output is assigned to refIdxL0.~~
- ~~Otherwise (modification_of_pic_nums_idc is equal to 3), the modification process for reference picture list RefPicList0 is finished.~~

~~When the current slice is a B slice and ref_pic_list_modification_flag_l1 is equal to 1, the following applies:~~

~~Let refIdxL1 be an index into the reference picture list RefPicList1. It is initially set equal to 0.~~

~~The corresponding syntax elements modification_of_pic_nums_idc are processed in the order they occur in the bitstream. For each of these syntax elements, the following applies.~~

~~If modification_of_pic_nums_idc is equal to 0 or equal to 1, the process specified in subclause 8.2.2.3.1 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.~~

~~Otherwise, if modification_of_pic_nums_idc is equal to 2, the process specified in subclause 8.2.2.3.2 is invoked with refIdxL1 as input, and the output is assigned to refIdxL1.~~

~~Otherwise (modification_of_pic_nums_idc is equal to 3), the modification process for reference picture list RefPicList1 is finished.~~

~~Modification process of reference picture lists for short-term reference pictures~~

~~Input to this process is an index refIdxLX (with X being 0 or 1).~~

~~Output of this process is an incremented index refIdxLX.~~

~~The variable picNumLXNoWrap is derived as follows.~~

~~If ref_pic_list_modification_idc is equal to 0, the following applies.~~

~~If the current reference picture list is RefPicList0, curRefPicSet is set to RefPicSetStCurr0.~~

~~Otherwise (the current reference picture list is RefPicList1), curRefPicSet is set to RefPicSetStCurr1.~~

~~Otherwise, if ref_pic_list_modification_idc is equal to 1, the following applies.~~

~~If the current reference picture list is RefPicList0, curRefPicSet is set to RefPicSetStCurr1.~~

~~Otherwise (the current reference picture list is RefPicList1), curRefPicSet is set to RefPicSetStCurr0.~~

~~Otherwise, if ref_pic_list_modification_idc is equal to 2, curRefPicSet is set to RefPicSetLtCurr.~~

~~The variable pocLX is derived as follows.~~

~~pocLX = curRefPicSet[ref_pic_set_idx] (8-9)~~

~~The following procedure is conducted to place the picture picR with PicOrderCnt(picR) equal to pocLX into the index position refIdxLX, shift the position of any other remaining pictures to later in the list, and increment the value of refIdxLX.~~

```
~~for( cIdx = num_ref_idx_lX_active_minus1 + 1; cIdx > refIdxLX; cIdx− − )~~
        ~~RefPicListX[ cIdx ] = RefPicListX[ cIdx − 1]~~
    ~~RefPicListX[ refIdxLX++ ] = pocLX~~
    ~~nIdx = refIdxLX                    (8-9)~~
    ~~for( cIdx = refIdxLX; cIdx <= num_ref_idx_lX_active_minus1 + 1; cIdx++ )~~
        ~~if( PicOrderCnt( RefPicListX[ cIdx ] ) != pocLX )~~
            ~~RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ]~~
```

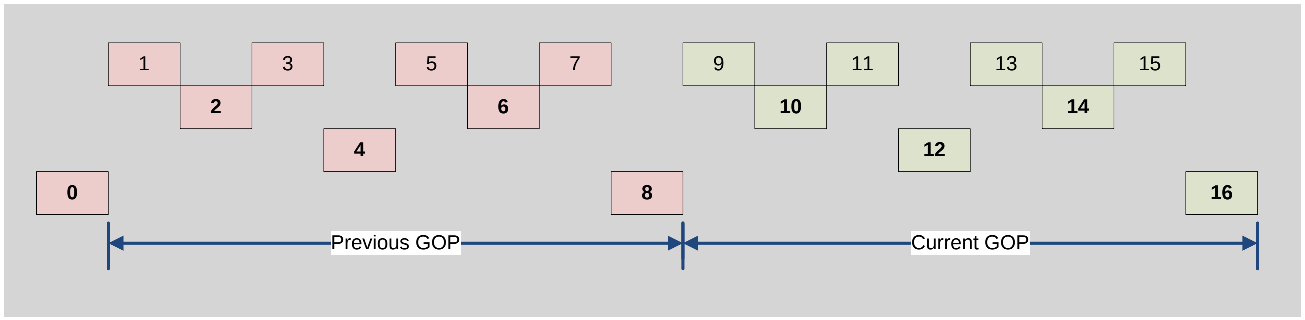
~~NOTE 2 – Within this pseudo-code procedure, the length of the list RefPicListX is temporarily made one element longer than the length needed for the final list. After the execution of this procedure, only elements 0 through num_ref_idx_lX_active_minus1 of the list need to be retained.~~

Benefits of proposed unification:

- Same modification method as LC
- More efficient signaling in many cases
- Easier to describe
- Overall reduction in WD text
  - Syntax: 4 lines
  - Semantics: ~10 lines + 1 table
  - Decoding process: ~35 lines

Encoder-only L0/L1 reordering

- Encoder-only L0/L1 reordering is implemented in HM5.0 to test the proposed change and facilitate bit counting
  - Maximize number of unique entries in L0 and L1



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

Simulation results (1)

Rate-distortion performance, reordered LC vs. default LC (HM5.0)

	Random Access HE			Random Access LC			Random Access HE-10		
	Y	U	V	Y	U	V	Y	U	V
Class A	0.0%	0.0%	-0.1%	0.0%	0.0%	0.0%	-0.0%	-0.3%	-0.3%
Class B	-0.2%	-0.1%	-0.1%	-0.1%	0.0%	-0.1%	-0.1%	-0.1%	-0.1%
Class C	-0.1%	0.0%	0.0%	0.0%	-0.1%	0.0%			
Class D	-0.1%	-0.2%	-0.1%	-0.1%	0.0%	-0.1%			
Class E									
Overall	-0.1%	-0.1%	-0.1%	-0.1%	0.0%	0.0%	-0.1%	-0.2%	-0.2%
	-0.1%	-0.1%	-0.1%	-0.1%	0.0%	0.0%	-0.1%	-0.2%	-0.1%
Class F	-0.1%	-0.1%	0.0%	-0.1%	-0.1%	-0.1%			

Per picture bit counting, proposed vs. current syntax for L0/L1 modification

POC	Modified lists		Existing ref_pic_list_modification()	Proposed ref_pic_list_modification()
	L0	L1		
16	{8, 6, 4, 0}	{8, 6, 4, 0}	2	2
12	{8, 6}	{16, 8}	2	2
10	{8, 6}	{12, 16}	2	2
9	{8, 12}	{10, 16}	26	10
11	{10, 8}	{12, 16}	2	2
14	{12, 10}	{16, 8}	15	8
13	{12, 8}	{14, 16}	2	2
15	{14, 12}	{16, 8}	15	8

We would like to thank Canon (JCTVC-H0679) for cross checking our RD results

Simulation results (2)

- G1036: Common conditions for reference picture marking and list construction proposals
  - Mandates bit counting for relevant syntax per RAP (random access period)
  - Defines additional testing conditions beyond those in G1200
  - Software branch HM-5.1-dev-ahg21
- Bit counting per RAP for ref_pic_list_modification() implemented on HM5.0
- Due to time constraint, only G1036 section 2.1 (G1200 RA) was used
- Compare to current syntax, the proposed syntax
  - Achieves 47% saving of signaling overhead for ref_pic_list_modification()

	Current syntax	Proposed syntax	% savings
RA-HE	6677	3566	46.7%
RA-LC	6677	3566	46.7%
RA-10	3399	1818	46.5%

Conclusion

- Unify L0/L1 list modification with LC modification
- Software changes to HM5.0
  - Implemented encoder-only list modification for L0/L1
  - Implemented bit counting for ref_pic_list_modification()
- The proposed unification is more efficient:
  - ~47% reduction in signaling bits
- WD text reduction and alignment
  - Syntax: 4 lines
  - Semantics: ~10 lines and 1 table
  - Decoding process: ~35 lines
- Suggest to adopt

Proposed decoding process for list initialization

1. RefPicSetCurrTempList0 is constructed from RefPicSetStCurr0, RefPicSetStCurr1 and RefPicSetLtCurr.

```
cIdx = 0
if (ref_pic_list_modification_flag_l0 == 0 )
    NumRpsCurrTempList0 = num_ref_idx_l0_active_minus1 + 1
else {
    NumRpsCurrTempList0 = NumRpsStCurr0 + NumRpsStCurr1 + NumRpsLtCurr
    if (NumRpsCurrTempList0 <= num_ref_idx_l0_active_minus1)
        NumRpsCurrTempList0 = num_ref_idx_l0_active_minus1+1
  }
while( cIdx < NumRpsCurrTempList0 )
{
   for( i=0; i < NumPocStCurr0 && cIdx < NumRpsCurrTempList0; cIdx++, i++ )
    RefPicSetCurrTempList0 [ cIdx ] = RefPicSetStCurr0[ i ]
   for( i=0;  i < NumPocStCurr1 && cIdx < NumRpsCurrTempList0; cIdx++, i++ )
    RefPicSetCurrTempList0 [ cIdx ] = RefPicSetStCurr1[ i ]
   for( i=0; i < NumPocLtCurr && cIdx < NumRpsCurrTempList0; cIdx++, i++ )
    RefPicSetCurrTempList0 [ cIdx ] = RefPicSetLtCurr[ i ]
}
```

2. If ref_pic_list_modification_flag_l0 is 0, the initial RefPicList0 is equivalent to RefPicSetCurrTempList0.

RefPicSetCurr0	Ref2	Ref1		
RefPicSetCurr1	Ref4	Ref5		
RefPicSetCurrTempL0 (default L0)	Ref2	Ref1	Ref4	Ref5
RefPicSetCurrTempL1 (default L1)	Ref4	Ref5	Ref2	Ref1

Additional results (1)

- Additional bit counting
  - Apply u(v) instead of te(v) on ref_pic_set_idx

Per picture bit counting, proposed (te(v) and u(v) for ref_pic_set_idx) vs. current syntax for L0/L1 modification

POC	Modified lists		Existing ref_pic_list_modification()	Proposed ref_pic_list_modification()	Proposed ref_pic_list_modification() with u(v) for ref_pic_set_idx
	L0	L1			
16	{8, 6, 4, 0}	{8, 6, 4, 0}	2	2	2
12	{8, 6}	{16, 8}	2	2	2
10	{8, 6}	{12, 16}	2	2	2
9	{8, 12}	{10, 16}	26	10	10
11	{10, 8}	{12, 16}	2	2	2
14	{12, 10}	{16, 8}	15	8	6
13	{12, 8}	{14, 16}	2	2	2
15	{14, 12}	{16, 8}	15	8	6

Additional Results (2)

- Additional bit counting for one RAP

	Current syntax	Proposed syntax	% savings	Proposed syntax with u(v)	% savings
RA-HE	6677	3566	46.7%	3174	52.5%
RA-LC	6677	3566	46.7%	3174	52.5%
RA-10	3399	1818	46.5%	1618	52.4%

Back up

L0/L1 size	# syntax current	# syntax proposed
1	3	1
2	3-5	2
3	3-7	3
4	3-9	4
5	3-11	5