

JCTVC-K0124: Reference picture list modification with truncation

Adarsh K. Ramasubramonian and
Ying Chen (Qualcomm)

Summary

- Add truncation functionality for RPLM entry loop
- Few changes as possible to syntax and semantics
 - Increase range of list entries by one
- Method 1
 - Swap first two entries
 - Bring a picture to the front of the list, and shift the entries
- Method 2
 - Simplified
 - Bring pictures to the beginning of the list, with duplication

Reference picture list (RPL) modification syntax

	Descriptor
if(slice_type == P slice_type == B) {	
ref_pic_list_modification_flag_l0	u(1)
if(ref_pic_list_modification_flag_l0 && NumPocTotalCurr > 1)	
for(i = 0; i <= num_ref_idx_l0_active_minus1 && (!i list_entry_l0[i - 1] < NumPocTotalCurr); i++)	
list_entry_l0[i]	u(v)
}	
if(slice_type == B) {	
ref_pic_list_modification_flag_l1	u(1)
if(ref_pic_list_modification_flag_l1 && NumPocTotalCurr > 1)	
for(i = 0; i <= num_ref_idx_l1_active_minus1 && (!i list_entry_l1[i - 1] < NumPocTotalCurr); i++)	
list_entry_l1[i]	u(v)
}	
}	

RPL modification semantics

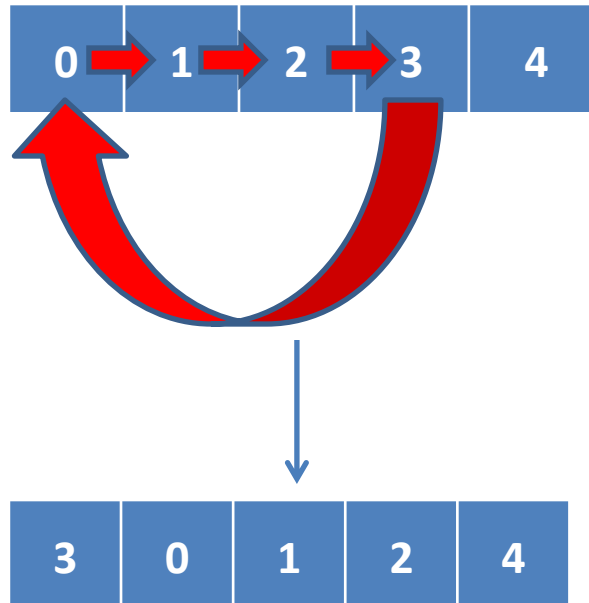
- **list_entry_IX**[i] (with X equal to 0 or 1) specifies the index of the reference picture in RefPicSetCurrTempListX to be placed at the current position of reference picture list LX (with X being 0 or 1). The length of the list_entry_IX[i] syntax element is $\text{Ceil}(\text{Log2}(\text{NumPocTotalCurr}+1))$ bits. The value of list_entry_IX[i] shall be in the range of 0 to NumPocTotalCurr~~—1~~, inclusive. If the syntax element list_entry_IX[i] is not present, it is inferred to be equal to 0.
- **NOTE** - When list_entry_IX[i] is equal to NumPocTotalCurr, the reference picture list modification syntax for RefPicListX ends.

Method 1 – RefPicIdxTempListX[]

Reference pictures used by current picture	A	B	C	D	E
RefPicIdxTempListX[]	0	1	2	3	4

num_ref_idx_IX_active_minus1 = 4

Syntax signalled: list_entry_IX[0] = 3, list_entry_IX[1] = 5



Method 1 – final list construction

3	0	1	2	4
---	---	---	---	---

RefPicIdxTempListX

D	A	B	C	E
---	---	---	---	---

Modified RefPicListX

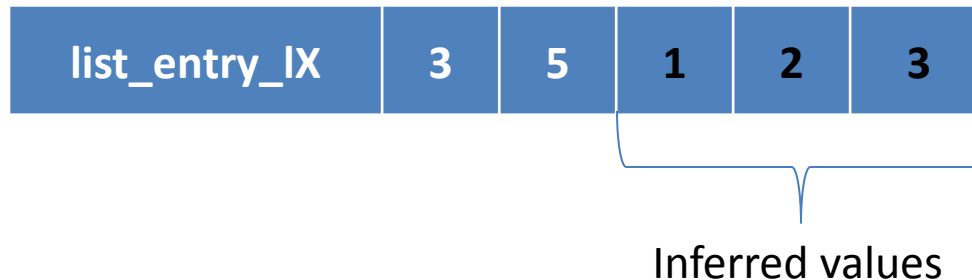
```
for( rIdx = 0; rIdx ≤ num_ref_idx_lX_active_minus1; rIdx++)           (8-9)
    RefPicListX[ rIdx ] = ref_pic_list_modification_flag_lX ?
        RefPicListTempX[ RefPicIdxTempListXlist_entry_lX[ rIdx ] ] : RefPicListTempX[ rIdx ]
```

Method 2 – list_entry_IX

Reference pictures used by current picture	A	B	C	D	E
--	---	---	---	---	---

num_ref_idx_IX_active_minus1 = 4

Syntax signalled: list_entry_IX[0] = 3, list_entry_IX[1] = 5



If the syntax element list_entry_IX[i] is not present, it is inferred to be equal to $(i == 0) ? 0 : (\text{list_entry_IX}[i - 1] + 1) \% \text{NumPocTotalCurr}$.

Method 2 – construction of list



.....

The list RefPicListX is constructed as follows:

```
for( rIdx = 0; rIdx ≤ num_ref_idx_lX_active_minus1; rIdx++)           (8-9)
  RefPicListX[ rIdx ] = ref_pic_list_modification_flag_lX ?
    RefPicListTempX[ list_entry_lX[ rIdx ] % NumPocTotalCurr ] ÷ RefPicListTempX[ rIdx ]
```


RPLM test conditions (J0513)

Bit-count of RPS-related syntax elements for random access test condition 2.8 (J0513)

	HEVC latest draft	Method 1	Method 2
Algorithm	6269	6709	6601
% change		+ 7.02%	+5.30%

Bit-count of RPS-related syntax elements for low-delay test condition 3.5 (J0513)

	HEVC latest draft	Method 1	Method 2
Algorithm	11243	10083	13563
% change		- 10.32%	+ 20.64%