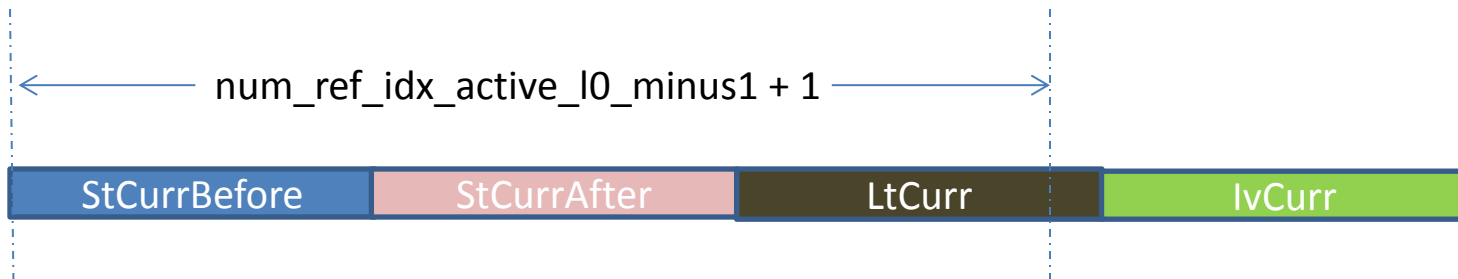


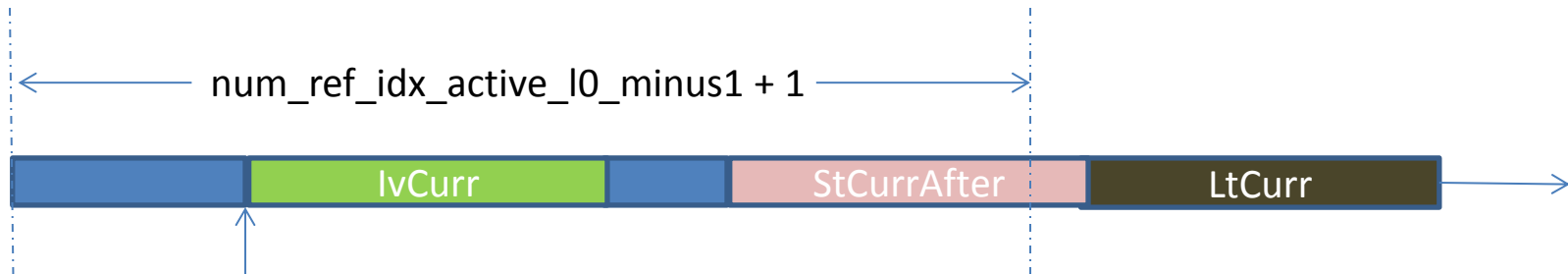
JCT3V-D0198: AHG7: Inter-layer reference pictures in reference picture list initialization

Adarsh K. Ramasubramonian, Ying
Chen, Ye-Kui Wang , Li Zhang
(Qualcomm)

Inter-view reference picture start position



Current initialization method



$\text{inter_view_ref_start_pos_plus1} - 1$

Proposed initialization method

Slice header extension changes

slice_header() {	Descriptor
first_slice_in_pic_flag	u(1)
...	
if(slice_header_extension_present_flag) { // should always be true in MV-HEVC	
slice_header_extension_length	ue(v)
if(slice_type != I_SLICE)	
inter_view_ref_start_position_L0_plus1	ue(v)
if(slice_type == B_SLICE)	
inter_view_ref_start_position_L1_plus1	ue(v)
...	
}	
byte_alignment()	
}	

```

if( !inter_view_ref_start_position_LX_plus1 )
    lvRefStartPosLX = NumPocStCurrBefore + NumPocStCurrAfter + NumPocLtCurr
else
    lvRefStartPosLX = inter_view_ref_start_position_LX_plus1 - 1

```

Decoding process for reference picture lists construction for list 0

When the slice is a P slice or a B slice, the variable NumRpsCurrTempList0 is set equal to $\text{Max}(\text{num_ref_idx_l0_active_minus1} + 1, \text{NumPocTotalCurr})$ and the list RefPicListTemp0 is constructed as follows:

```
cldx = 0
while( cldx < NumRpsCurrTempList0 - NumPocIvCurr ) {
    for( i = 0; i < NumPocStCurrBefore && cldx < NumRpsCurrTempList0; cldx++, i++ )
        RefPicListTemp0[ cldx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocStCurrAfter && cldx < NumRpsCurrTempList0; cldx++, i++ )
        RefPicListTemp0[ cldx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && cldx < NumRpsCurrTempList0; cldx++, i++ )
        RefPicListTemp0[ cldx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumIvCurr && rldx < NumRpsCurrTempList0; rldx++, i++ )
    RefPicListTemp0[ rldx ] = RefPicSetIvCurr[ i ]
}
// Shift the pictures in the list to create space for inter-view reference pictures, if necessary
for( cldx = NumRpsCurrTempList0 - 1; cldx >= IvRefStartPosL0 + NumPocIvCurr; cldx -- )
    RefPicListTemp0[ cldx ] = RefPicListTemp0[ cldx - NumPocIvCurr ]

// Include the inter-view reference pictures
for( i=0, cldx = IvRefStartPosL0; i < NumPocIvCurr; cldx++, i++ )
    RefPicListTemp0[ cldx ] = RefPicSetIvCurr[ i ]
```

The list RefPicList0 is constructed as follows:

```
for( cldx = 0; cldx ≤ num_ref_idx_l0_active_minus1; cldx++ )
    RefPicList0[ cldx ] = ref_pic_list_modification_flag_l0 ? RefPicListTemp0[ list_entry_l0[ cldx ] ]
    : RefPicListTemp0[ cldx ]
```

(F-26)

Decoding process for reference picture lists construction for list 1

When the slice is a B slice, the variable NumRpsCurrTempList1 is set equal to $\text{Max}(\text{num_ref_idx_l1_active_minus1} + 1, \text{NumPocTotalCurr})$ and the list RefPicListTemp1 is constructed as follows:

$\text{cldx} = 0$

```

while( cldx < NumRpsCurrTempList1 - NumPocIvCurr ) {
    for( i = 0; i < NumPocStCurrAfter && cldx < NumRpsCurrTempList1; cldx++, i++ )
        RefPicListTemp1[ cldx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocStCurrBefore && cldx < NumRpsCurrTempList1; cldx++, i++ )
        RefPicListTemp1[ cldx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocLtCurr && cldx < NumRpsCurrTempList1; cldx++, i++ )
        RefPicListTemp1[ cldx ] = RefPicSetLtCurr[ i ]
    for( i = 0; i < NumIvCurr && rldx < NumRpsCurrTempList1; rldx++, i++ )
        RefPicListTemp0[ rldx ] = RefPicSetIvCurr[ i ]
}

```

```
// Shift the pictures in the list to create space for inter-view reference pictures, if necessary
```

```
for( cldx = NumRpsCurrTempList1 - 1; cldx >= lvRefStartPosL1 + NumPoclvCurr; cldx - - )
```

```
RefPicListTemp1[ cldx ] = RefPicListTemp1[ cldx - NumPocIvCurr ]
```

```
// Include the inter-view reference pictures
```

```
for( i=0, cldx = lvRefStartPosL1; i < NumPoclvCurr; cldx++, i++ )
```

```
RefPicListTemp1[ cldx ] = RefPicSetIvCurr[ i ]
```

The list RefPicList1 is constructed as follows:

```
for( cldx = 0; cldx ≤ num_ref_idx_l1_active_minus1; cldx++)
```

[illegible]

Results

- Bit-count comparison for sending RPLM commands and the inter-view reference start positions for base + 2 views

# frames	Anchor (HTM 6.0)	Proposed method (List 0 and List 1)	M0081/D0057 Option 1 (Only List 0)	M0081/D0057 Option 2 (List 0 and List 1)
300 frames	4746	2965	2424	7162
		- 37.5%	- 48.9%	+ 50.9%
500 frames	7962	4949	4098	12036
		- 37.8%	- 48.5%	+ 51.2%

Thank you