

## JCTVC-Q0107 SHVC HLS: On Picture Level Resampling Phase Filter Selection



invention | collaboration | contribution

Yan Ye, Yuwen He, Yong He  
InterDigital Communications, Inc.

17<sup>th</sup> JCT-VC meeting, Mar/Apr  
2014

# Resampling phase related syntax in SHVC draft 5

- Flags used to derive resampling filter phase are not harmonized

	cross_layer_phase_alignment_flag	vert_phase_position_enable_flag	vert_phase_position_flag
Syntax location	VPS	SPS extension	Slice header
Syntax purpose	Top-left or center aligned DS filter	Field to frame scalability	

- Scaled reference offsets and vert\_phase\_position\_enable\_flag are tied together in SPS extension, although not necessarily related

sps_multilayer_extension( ) {	Descriptor
inter_view_mv_vert_constraint_flag	u(1)
num_scaled_ref_layer_offsets	ue(v)
for( i = 0; i < num_scaled_ref_layer_offsets; i++) {	
scaled_ref_layer_id[ i ]	u(6)
scaled_ref_layer_left_offset[ scaled_ref_layer_id[ i ] ]	se(v)
scaled_ref_layer_top_offset[ scaled_ref_layer_id[ i ] ]	se(v)
scaled_ref_layer_right_offset[ scaled_ref_layer_id[ i ] ]	se(v)
scaled_ref_layer_bottom_offset[ scaled_ref_layer_id[ i ] ]	se(v)
vert_phase_position_enable_flag[ scaled_ref_layer_id[ i ] ]	u(1)
}	
}	

- Can't combine vert\_phase\_position\_flag with cross\_layer\_phase\_alignment\_flag
- Small efficiency loss for non-default chroma sample location (JCTVC-P0164)

# Proposed solution

## Resampling filter phase selection in SHVC draft 5:

### 1. Derive phaseX and phaseY (in H.6.2)

$\text{phaseX} = (\text{cldx} == 0) ? (\text{cross\_layer\_phase\_alignment\_flag} \ll 1) : \text{cross\_layer\_phase\_alignment\_flag}$  (H-5)

$\text{phaseY} = \text{VertPhasePositionAdjustFlag} ? (\text{VertPhasePositionFlag} \ll 2) :$

$( (\text{cldx} == 0) ? (\text{cross\_layer\_phase\_alignment\_flag} \ll 1) : \text{cross\_layer\_phase\_alignment\_flag} + 1)$  (H-6)

### 2. Calculate xRef16 and yRef16 (in H.6.2)

$\text{addX} = (\text{ScaleFactorX} * \text{phaseX} + 2) \gg 2$  (H-7)

$\text{addY} = (\text{ScaleFactorY} * \text{phaseY} + 2) \gg 2$  (H-8)

$\text{xRef16} = (( (\text{xP} - \text{offsetX}) * \text{ScaleFactorX} + \text{addX} + (1 \ll 11)) \gg 12) - (\text{phaseX} \ll 2)$  (H-9)

$\text{yRef16} = (( (\text{yP} - \text{offsetY}) * \text{ScaleFactorY} + \text{addY} + (1 \ll 11)) \gg 12) - (\text{phaseY} \ll 2)$  (H-10)

### 3. Use the last 4 LSB from xRef16 and yRef16 to choose horizontal and vertical resampling filter phases (in H.8.1.4.1.3)

## Proposed solution:

Directly signal the values of phaseXLuma, phaseXChroma, phaseYLuma, and phaseYChroma in (H-5) and (H-6)

Provide the flexibility for the encoder to decide what phases are used for specific application

# Comparing current syntax and proposed syntax

	Current syntax			Proposed syntax			
	Cross_layer_phase_alignment_flag (VPS)	vert_phase_position_enable_flag (SPS extension)	vert_phase_position_flag (slice header)	phaseXLuma	phaseYLuma	phaseXChroma	phaseYChroma
Top-left aligned DS filter	0	0	-	0	0	0	1
Center aligned DS filter	1	0	-	2	2	1	2
top field to frame	0	1	0	0	0	0	0
bottom field to frame	0	1	1	0	4	0	4
chroma sample location type-2				0	0	0	0

CTC

# Option 1: signaling one phase set in PPS extension

pic_parameter_set_rbsp( ) {	Descriptor
...	
pps_extension_flag	u(1)
if( pps_extension_flag ) {	
for ( i = 0; i < 8; i++ )	
pps_extension_type_flag[ i ]	u(1)
if( pps_extension_type_flag[ 0 ] ) {	
poc_reset_info_present_flag	u(1)
<b>resample_phase_set_present_flag</b>	<b>u(1)</b>
if ( resample_phase_set_present_flag ) {	
<b>resample_phase_x_luma</b>	<b>u(3)</b>
<b>resample_phase_y_luma</b>	<b>u(3)</b>
<b>resample_phase_x_chroma</b>	<b>u(3)</b>
<b>resample_phase_y_chroma</b>	<b>u(3)</b>
<b>}</b>	
<b>}</b>	
...	
}	
rbbsp_trailing_bits( )	
}	

# Option 1: semantics and decoding process

**resample\_phase\_set\_present\_flag** equal to 1 specifies that **resample\_phase\_x\_luma**, **resample\_phase\_y\_luma**, **resample\_phase\_x\_chroma**, and **resample\_phase\_y\_chroma** are present in the PPS RBSP syntax structure. **resample\_phase\_set\_present\_flag** equal to 0 specifies that **resample\_phase\_x\_luma**, **resample\_phase\_y\_luma**, **resample\_phase\_x\_chroma**, and **resample\_phase\_y\_chroma** are not present in the PPS RBSP syntax structure.

**resample\_phase\_x\_luma** specifies the horizontal resampling phase adjustment value used to select the luma resampling filter in the horizontal direction. When **resample\_phase\_x\_luma** is not present, it is inferred to be equal to 0.

**resample\_phase\_y\_luma** specifies the vertical resampling phase adjustment value used to select the luma resampling filter in the vertical direction. When **resample\_phase\_y\_luma** is not present, it is inferred to be equal to 0.

**resample\_phase\_x\_chroma** specifies the horizontal resampling phase adjustment value used to select the chroma resampling filter in the horizontal direction. When **resample\_phase\_x\_chroma** is not present, it is inferred to be equal to 0.

**resample\_phase\_y\_chroma** specifies the vertical resampling phase adjustment value used to select the chroma resampling filter in the vertical direction. When **resample\_phase\_y\_chroma** is not present, it is inferred to be equal to 1.

$$\text{phaseX} = (\text{cldx} == 0) ? \text{resample\_phase\_x\_luma} : \text{resample\_phase\_x\_chroma} \quad (\text{H-5})$$
$$\text{phaseY} = (\text{cldx} == 0) ? \text{resample\_phase\_y\_luma} : \text{resample\_phase\_y\_chroma} \quad (\text{H-6})$$

## Option 2: signaling one or more phase sets

pic_parameter_set_rbsp( ) {	Descriptor
...	
pps_extension_flag	u(1)
if( pps_extension_flag ) {	
for ( i = 0; i < 8; i++ )	
pps_extension_type_flag[ i ]	u(1)
if( pps_extension_type_flag[ 0 ] ) {	
poc_reset_info_present_flag	u(1)
<b>num_resample_phase_sets</b>	<b>ue(v)</b>
for ( i = 0; i < num_resample_phase_sets; i++ ) {	
<b>resample_phase_x_luma[ i ]</b>	<b>u(3)</b>
<b>resample_phase_y_luma[ i ]</b>	<b>u(3)</b>
<b>resample_phase_x_chroma[ i ]</b>	<b>u(3)</b>
<b>resample_phase_y_chroma[ i ]</b>	<b>u(3)</b>
<b>}</b>	
<b>}</b>	
...	
}	
rbsp_trailing_bits( )	
}	

## Option 2: slice header syntax

slice_segment_header( ) {	Descriptor
...	
if( nuh_layer_id > 0 && !all_ref_layers_active_flag && NumDirectRefLayers[ nuh_layer_id ] > 0 ) {	
inter_layer_pred_enabled_flag	u(1)
if( inter_layer_pred_enabled_flag && NumDirectRefLayers[ nuh_layer_id ] > 1 ) {	
if( !max_one_active_ref_layer_flag )	
num_inter_layer_ref_pics_minus1	u(v)
if( NumActiveRefLayerPics != NumDirectRefLayers[ nuh_layer_id ] )	
for( i = 0; i < NumActiveRefLayerPics; i++ )	
inter_layer_pred_layer_idc[ i ]	u(v)
}	
}	
for( i = 0; i < NumActiveRefLayerPics; i++ )	
if ( vert_phase_position_enable_flag[ RefPicLayerId[ i ] ] )	
vert_phase_position_flag[ RefPicLayerId[ i ] ]	u(1)
if( nuh_layer_id > 0 && num_resample_phase_sets > 1 ) {	
for( i = 0; i < NumActiveRefLayerPics; i++ )	
if( ResampleFromRefLayerFlag[ RefPicLayerId[ i ] ] )	
phase_set_idx[ RefPicLayerId[ i ] ]	u(v)
}	
...	
}	



## Option 2: semantics

**num\_resample\_phase\_sets** specifies the number of resample phase set syntax elements: `resample_phase_x_luma[ i ]`, `resample_phase_y_luma[ i ]`, `resample_phase_x_chroma[ i ]`, `resample_phase_y_chroma[ i ]`.

**resample\_phase\_x\_luma[ i ]** specifies the i-th horizontal resampling phase adjustment value used to select the luma resampling filter in the horizontal direction. When `resample_phase_x_luma[ i ]` is not present, it is inferred to be equal to 0.

**resample\_phase\_y\_luma[ i ]** specifies the i-th vertical resampling phase adjustment value used to select the luma resampling filter in the vertical direction. When `resample_phase_y_luma[ i ]` is not present, it is inferred to be equal to 0.

**resample\_phase\_x\_chroma[ i ]** specifies the i-th horizontal resampling phase adjustment value used to select the chroma resampling filter in the horizontal direction. When `resample_phase_x_chroma[ i ]` is not present, it is inferred to be equal to 0.

**resample\_phase\_y\_chroma[ i ]** specifies the i-th vertical resampling phase adjustment value used to select the chroma resampling filter in the vertical direction. When `resample_phase_y_chroma[ i ]` is not present, it is inferred to be equal to 1.

**phase\_set\_idx[ i ]** specifies the phase set index used to resample the reference layer picture with `nuh_layer_id` equal to `i` for decoding of the current slice. When `phase_set_idx` is not present, it is inferred to be equal to 0. The value of `phase_set_idx` shall be in the range of 0 and `num_resample_phase_sets - 1`, inclusive.

## Option 2: decoding process

### Derivation of ResampleFromRefLayerFlag[refLayerId] (IsSpatialEnhLayer() in SHM5.0\*)

```
for( i = 0; i < NumActiveRefLayerPics; i++ ) {  
    refLayerId = RefPicLayerId[ i ]  
    rIPic is the decoded reference layer picture  
    refLayerPicWidthInSamplesY is set to the width of rIPic in units of luma samples  
    refLayerPicHeightInSamplesY is set to the height of rIPic in units of luma samples  
    refLayerBitDepthY is set to BitDepthY of rIPic  
    refLayerBitDepthC is set to BitDepthC of rIPic  
    if (pic_width_in_luma_samples == refLayerPicWidthInSamplesY &&  
        pic_height_in_luma_samples == refLayerPicHeightInSamplesY &&  
        scaled_ref_layer_left_offset[ refLayerId ] == 0 && scaled_ref_layer_left_offset[ refLayerId ] == 0 &&  
        scaled_ref_layer_left_offset[ refLayerId ] == 0 && scaled_ref_layer_left_offset[ refLayerId ] == 0 &&  
        refLayerBitDepthY == BitDepthY && refLayerBitDepthC == BitDepthC )  
        ResampleFromRefLayerFlag[refLayerId] = 0  
    else  
        ResampleFromRefLayerFlag[refLayerId] = 1  
}
```

### The process in H.8.1.4 is modified as follows:

VertPhasePositionFlag = vert\_phase\_position\_flag[ rLId ] ——— (H-20)  
PhaseSetIdx = phase\_set\_idx[ rLId ] (H-20)

### The process in H.6.2 is modified as follows:

phaseX = ( cldx == 0 ) ? resample\_phase\_x\_luma[ PhaseSetIdx ] : resample\_phase\_x\_chroma[ PhaseSetIdx ] (H-5)  
phaseY = ( cldx == 0 ) ? resample\_phase\_y\_luma[ PhaseSetIdx ] : resample\_phase\_y\_chroma[ PhaseSetIdx ] (H-6)

# Conclusion

- SHVC draft 5 derives the resampling filter phases from a number of flags
- Propose to signal phaseXLuma, phaseYLuma, phaseXChroma, and PhaseYChroma directly
- Two options are proposed
- It is suggested to consider Option 2 as the more flexible and more efficient solution