JCTVC-D202: 1:2 Spatial Scalability Support for HEVC

Danny Hong, Jill Boyce, Alexandros Eleftheriadis

Vidyo

JCTVC-D202: 1:2 Spatial Scalability Support for HEVC

- Extend HEVC to include SVC 1:2 scalability tools
- Benefits:
  - Significant coding gain vs. simulcast
  - Clean design

Introduction

- Using scalable codec, a single bitstream can be scaled, without transcoding, to support various endpoints
- Simulcast results in significant bit rate overhead
- Incorporating scalability tools in the initial phase of HEVC results in a cleaner design, avoiding the need to retrofit the system design with ugly backwards-compatible patches, such as the NAL unit header SVC extension and the prefix NAL unit.

SVC has been embraced by the videoconferencing industry

Vidyo™ Personal Telepresence

- Announcements of SVC support by
  - Vidyo
  - Google
  - HP
  - Microsoft
  - Hitachi
  - Polycom
  - Radvision
  - …

Limited scalability support

- **SVC supported a wide range of scalability capabilities, but only a small subset of features have seen market adoption**
  - Temporal scalability
  - 1:2 Spatial scalability
- **Restricting proposal allows for quicker development, fewer pages of spec text**
  - Allows for inclusion in first phase of standard development
  - Temporal scalability already supported in HEVC WD
  - 1:2 spatial scalability easier to incorporate to into HEVC design than 1.5 and arbitrary ratios

Propose to adopt the SVC inter-layer prediction tools

- SVC inter-layer prediction tools
  - Inter-layer intra prediction
  - Inter-layer motion prediction
  - Inter-layer residual prediction
- Syntactic elements needed for prediction units
  - base_mode_flag
  - motion_prediction_flag_l0 and motion_prediction_flag_l1
  - residual_prediction_flag
- SVC upsampling method used

Syntax changes: NAL unit header

nal_unit(NumBytesInNALunit) {	**Descriptor**
…	
if(nal_unit_type == 1 | | nal_unit_type == 5) {	
temporal_id	u(3)
output_flag	u(1)
non_base_layer_flag	u(1)
reserved_zero_3bits	u(3)
if(non_base_layer_flag) {	
no_inter_layer_pred_flag	u(1)
dependency_id_minus1	u(2)
discardable_flag	u(1)
reserved_non_base_layer_zero_4bits	u(4)
nalUnitHeaderBytes += 1	
}	
nalUnitHeaderBytes += 1	
}	
…	
}	

Syntax changes: slice header

Vidyo™ Personal Telepresence

slice_header( ) {	Descriptor
…	
if(deblocking_filter_control_present_flag) {	
…	
}	
if(!no_inter_layer_pred_flag) {	
adaptive_base_mode_flag	u(1)
if(!adaptive_base_mode_flag)	
default_base_mode_flag	u(1)
if(!default_base_mode_flag) {	
adaptive_motion_prediction_flag	u(1)
if(!adaptive_motion_prediction_flag)	
default_motion_prediction_flag	u(1)
}	
adaptive_residual_prediction_flag	u(1)
if(!adaptive_residual_prediction_flag)	
default_residual_prediction_flag	u(1)
}	
}	

Syntax changes: coding unit

coding_unit(x0, y0, log2CUSize) {	**Descriptor**
if(slice_type != I)	
skip_flag[x0 **][** y0 **]**	u(1) | ae(v)
if(skip_flag[x0][y0])	
prediction_unit(x0, y0, log2CUSize, log2CUSize)	
else {	
if(NumMergeCandidates > 0)	
merge_flag[x0 **][** y0 **]**	u(1) | ae(v)
if(merge_flag[x0][y0]) {	
...	
} else {	
if(adaptive_base_mode_flag)	
base_mode_flag	u(1) | ae(v)
if(base_mode_flag) {	
prediction_unit(x0, y0, log2CUSize, log2CUSize)	
transform_tree(x0, y0, log2CUSize, 0, 0)	
} else {	
if(slice_type != I)	
pred_mode	u(1) | ae(v)
x1 = x0 + ((1 << log2CUSize) >> 1)	
y1 = y0 + ((1 << log2CUSize) >> 1)	
if(PredMode == MODE_INTRA) {	
…	ue(1) | ae(v)
} else if(PredMode = = MODE_INTER) {	
…	
} else { /* MODE_DIRECT */	
…	
}	
} /* !base_mode_flag */	
}	
}	
}	

<table>
<tr><th>prediction_unit(x0, y0, log2PUWidth, log2PUHeight) {</th><th>Descriptor</th></tr>
<tr><td>if(skip_flag[x0][y0]) {</td><td></td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>} else {</td><td></td></tr>
<tr><td>if(!base_mode_flag) {</td><td></td></tr>
<tr><td>if(PredMode == MODE_INTRA) {</td><td></td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>} else { /* MODE_MERGE, MODE_DIRECT, MODE_INTER */</td><td></td></tr>
<tr><td>if(merge_flag[x0][y0] && NumMergeCandidates > 1) {</td><td></td></tr>
<tr><td>**merge_left_flag[** x0 **][** y0 **]**</td><td>u(1) | ae(v)</td></tr>
<tr><td>} else if(PredMode == MODE_DIRECT) {</td><td></td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>} else {</td><td></td></tr>
<tr><td>if(slice_type == B)</td><td></td></tr>
<tr><td>**inter_pred_idc[** x0 **][** y0 **]**</td><td>ue(v) | ae(v)</td></tr>
<tr><td>if(inter_pred_idc[x0][y0] != Pred_L1) {</td><td></td></tr>
<tr><td>if(adaptive_motion_prediction_flag)</td><td></td></tr>
<tr><td>**motion_prediction_flag_l0[i]**</td><td>ue(1) | ae(v)</td></tr>
<tr><td>if(!motion_prediction_flag_l0[i]) {</td><td></td></tr>
<tr><td>… /* ref_idx_l0, mvd_l0, mvp_idx_l0 */</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>else { /* mvd_l0 */ }</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>if(inter_pred_idc[x0][y0] != Pred_L0) {</td><td></td></tr>
<tr><td>if(adaptive_motion_prediction_flag)</td><td></td></tr>
<tr><td>**motion_prediction_flag_l1[i]**</td><td>ue(1) | ae(v)</td></tr>
<tr><td>if(!motion_prediction_flag_l1[i]) {</td><td></td></tr>
<tr><td>… /* ref_idx_l1, mvd_l1, mvp_idx_l1 */</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>else { /* mvd_l1 */ }</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>}</td><td></td></tr>
<tr><td>} /* !base_mode_flag */</td><td></td></tr>
<tr><td>if(adaptive_residual_prediction_flag && slice_type != I && (base_mode_flag || PredMode != MODE_INTRA))</td><td></td></tr>
<tr><td>**residual_prediction_flag**</td><td>ue(1) | ae(v)</td></tr>
<tr><td>} /* !skip_flag[x0][y0] */</td><td></td></tr>
<tr><td>}</td><td></td></tr>
</table>

Experimental Conditions

Vidyo™ Personal Telepresence

- TMuC 0.9 was modified
  - Support 2 spatial layers (one base layer and one enhancement layer)
  - Intra only
  - 1:2 spatial scalability
  - Software available, uploaded with contribution
- All test sequences of the 5 classes
- Both high-efficiency and low-complexity settings
- QP offset of 0 and 2 in enhancement layer (vs. the base layer)
- Syntax proposal applicable to all picture types, but implementation and experimental results only available for Intra coding

Experimental Results (QP offset = 0)

- 2 layer intra spatial scalable coding vs. simulcast (comparing enhancement layer PSNR vs. total bit rate)

	Intra			Intra LoCo		
	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate
Class A	-19.4	-21.2	-21.9	-17.8	-20.0	-20.2
Class B	-15.2	-18.1	-17.6	-13.1	-15.0	-14.9
Class C	-7.9	-10.5	-9.7	-7.1	-8.8	-8.6
Class D	-6.5	-8.4	-7.6	-6.0	-8.1	-7.8
Class E	-15.3	-17.0	-16.6	-13.3	-14.5	-14.5
All	**-12.1**	**-14.4**	**-14.0**	**-10.7**	**-12.6**	**-12.4**
Max	**-25.4**	**-28.1**	**-29.6**	**-21.7**	**-25.4**	**-26.3**

Experimental Result (QP offset = 2)

- 2 layer intra spatial scalable coding vs. simulcast (comparing enhancement layer PSNR vs. total bit rate)

	Intra			Intra LoCo		
	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate
Class A	-22.8	-25.1	-26.2	-21.9	-25.1	-25.8
Class B	-18.4	-21.9	-21.7	-16.7	-19.6	-19.7
Class C	-10.3	-13.1	-12.3	-9.3	-11.8	-11.6
Class D	-8.4	-10.3	-9.6	-7.7	-10.8	-10.5
Class E	-18.7	-21.8	-21.2	-16.9	-20.5	-20.2
All	**-14.9**	**-17.7**	**-17.3**	**-13.7**	**-16.7**	**-16.6**
Max	**-29.8**	**-33.7**	**-35.4**	**-27.4**	**-31.9**	**-32.3**

Conclusions

- SVC is widely adopted and accepted by the videoconferencing industry
- Scalable coding yields significant coding gains compared to simulcast
- Incorporating scalability features in the first phase ensures a clean scalability design and eliminates the need to retrofit the design later to add scalability support
- Limiting features allows for simpler design
- Recommend for group to define experimental conditions for spatial scalability

Vidyo
Personal Telepresence

Thank you