



JCTVC-J0245: SEI Message for profile and level signaling for temporal scalability and extensions

Jill Boyce, Wonkap Jang, and Danny Hong
Vidyo



Introduction

- **SEI message proposed to optionally indicate sub-bitstream profiles and levels**
 - Applicable to temporal sub-layers in current HEVC base specification
 - Also applicable to views/layers in extension
- **Based upon JCTC-I0231**
 - Updated to consider new profile indicators adopted in JCTVC-I0499
- **In extension, define pixel throughput level limit to apply only to individual layer**
- **Mental cross-check provided in JCTVC-J0507**

SEI Message Proposal

- **Current HEVC design defines sub-bitstream extraction process**
 - Input is target temporal_id value, output is compliant sub-bitstream
 - Extension will likely define process for input target view_id, depth_flag, etc. values
 - No means to signal profile & level of sub-bitstream separately
- **SEI message allows signaling of profile and/or level for sub-bitstreams**
 - Loop through possible values of temporal_id for base specification, and also layer_id in extensions
 - Separate presence flags for profile and level
 - Profile flag indicates presence of sub-bitstream profile and reserved_zero_8bits
 - Level flag indicates present of sub-bitstream level
 - Separate flags based on assumption that level more likely to differ than profile for temporal-based sub-bitstreams

Discussion of Alternatives

- **Profile and level indicators for temporal sub-bitstreams could alternatively be sent in VPS or SPS**
- **Useful for a middle box to have initial profile & level information available upfront for all layers and temporal sub-layers when processing a bitstream**
 - If sent in SPS, not all info available when middle box begins processing sequence
- **Placing any profile & level information in the VPS constrains flexibility of adaptation of individual layers within a coded video sequence**
 - Changing profile or level of ANY layer requires intra-coded random access picture (IDR, etc.) for ALL layers
 - Could have a “worst case” level in the VPS, but no concept of a “worst case” profile

Proposed SEI message syntax

sub_bitstream_profile_level_info() {	Descriptor
video_parameter_set_id	ue(v)
for (j = 0; j <= max_temporal_layers_minus1; j ++) {	
sub_bitstream_profile_info_present_flag[0][j]	u(1)
if (sub_bitstream_profile_info_present_flag[0][j]) {	
sub_bitstream_profile_space	u(3)
sub_bitstream_profile_idc	u(5)
sub_bitstream_constraint_flags	u(16)
for(k = 0; k < 32; k++)	
sub_bitstream_profile_comptability_flag[0][j][k]	u(1)
}	
sub_bitstream_level_info_present_flag[0][j]	u(1)
if (sub_bitstream_level_info_present_flag[0][j]) {	
sub_bitstream_level_idc[0][j]	u(8)
}	
}	
}	

Proposed SEI message syntax for extensions

sub_bitstream_profile_level_info() {	Descriptor
sub_bitstream_video_parameter_set_id	ue(v)
for (i = 0; i <= max_layers_minus1; i++) {	
for (j = 0; j <= max_temporal_layers_minus1; j ++) {	
sub_bitstream_profile_info_present_flag[i][j]	u(1)
if (sub_bitstream_profile_info_present_flag[i][j]) {	
sub_bitstream_profile_space[i][j]	u(3)
sub_bitstream_profile_idc[i][j]	u(5)
sub_bitstream_constraint_flags[i][j]	u(16)
for(k = 0; k < 32; k++)	
sub_bitstream_profile_compatibility_flag[i][j][k]	u(1)
}	
sub_bitstream_level_info_present_flag[i][j]	u(1)
if (sub_bitstream_level_info_present_flag[i][j])	
sub_bitstream_level_idc[i][j]	u(8)
}	
}	
}	
}	

Background: SVC, MVC SEI messages

- **SVC scalability information SEI message**

- Provides number of layers present in the coded video sequence
- Loops through layers
 - Provides a mapping of a layer id value to the priority, dependency, quality and temporal id values
 - Optionally provides profile and level information for each layer_id value

- **MVC view scalability information SEI message**

- Provides number of views present in the coded video sequence
- Optionally provides profile and level information for the sub-bitstream associated with a target view representation.

Background: SVC, MVC SEI messages

scalability_info(payloadSize) {	Descriptor
temporal_id_nesting_flag	u(1)
priority_layer_info_present_flag	u(1)
priority_id_setting_flag	u(1)
num_layers_minus1	ue(v)
for(i = 0; i <= num_layers_minus1; i++) {	
layer_id[i]	ue(v)
priority_id[i]	u(6)
discardable_flag[i]	u(1)
dependency_id[i]	u(3)
quality_id[i]	u(4)
temporal_id[i]	u(3)
...	
profile_level_info_present_flag[i]	u(1)
...	
if(profile_level_info_present_flag[i])	
layer_profile_level_idc[i]	u(24)
if(bitrate_info_present_flag[i]) {	
avg_bitrate[i]	u(16)
max_bitrate_layer[i]	u(16)
max_bitrate_layer_representation[i]	u(16)
max_bitrate_calc_window[i]	u(16)
...	

view_scalability_info(payloadSize) {	Descriptor
num_operation_points_minus1	ue(v)
for(i = 0; i <= num_operation_points_minus1; i++) {	
operation_point_id[i]	ue(v)
priority_id[i]	u(5)
temporal_id[i]	u(3)
num_target_output_views_minus1[i]	ue(v)
for(j = 0; j <= num_target_output_views_minus1[i]; j++)	
view_id[i][j]	ue(v)
profile_level_info_present_flag[i]	u(1)
bitrate_info_present_flag[i]	u(1)
frm_rate_info_present_flag[i]	u(1)
if(!num_target_output_views_minus1[i])	
view_dependency_info_present_flag[i]	u(1)
parameter_sets_info_present_flag[i]	u(1)
bitstream_restriction_info_present_flag[i]	u(1)
if (profile_level_info_present_flag[i])	
op_profile_level_idc[i]	u(24)
if(bitrate_info_present_flag[i]) {	
avg_bitrate[i]	u(16)
max_bitrate[i]	u(16)
max_bitrate_calc_window[i]	u(16)
}	
...	

Background: AVC, SVC, MVC level constraints



- **Constraint on maximum macroblock or pixel throughput considered**
 - All equations simplified from specification text, assuming fixed frame rate
- **AVC: $\text{MaxMBPS} \geq \text{PicSizeInMbs} * \text{FrameRate}$**
- **SVC: $\text{MaxMBPS} \geq \text{svcPicSizeInMbs} * \text{FrameRate}$**
 - where svcPicSizeInMbs is based on the number of layers and the picture size of the active layer and its reference layers
- **MVC: $\text{MaxMBPS} \geq (\text{NumViews} / 2) * \text{PicSizeInMbs} * \text{FrameRate}$**
 - where NumViews refers to the number of views required for decoding the target output view
- **Base HEVC: $\text{MaxLumaPR} \geq \text{PicSizeLuma} * \text{FrameRate}$**
 - Limit placed on pixel rate, rather than macroblock rate

Discussion on MVC level constraint

- **MVC's macroblock throughput limit is based upon the number of active views**
- **Multiple view layers whose SPS parameter values are otherwise identical, (e.g. with the same image resolution) differ in their level indicator value**
 - They must refer to different SPS ids, and separate PPS ids
 - Inefficient:
 - Multiple nearly identical sequence and picture parameter sets must be transmitted
 - Increase in the number of bits in each slice header to refer different picture parameter set id values

Proposed level constraint in extension

- **Level constraint in SPS refers only to the individual layer, and not the corresponding sub-bitstream**
 - Allows multiple view layers to refer to the same SPS
- **Use proposed SEI message to send sub-bitstream levels**
- **Rename `layer_idc` as `level_layer_idc`**

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>profile_idc</code>	<code>u(8)</code>
<code>reserved_zero_8bits /* equal to 0 */</code>	<code>u(8)</code>
<code>layer_level_idc</code>	<code>u(8)</code>
<code>seq_parameter_set_id</code>	<code>ue(v)</code>
<code>...</code>	

`layer_level_idc` indicates the level to which layer conforms.

Example

- **3-view multiview bitstream – Layers 0, 1, 2**
- **3 temporal sub-layers, same frame rate for each view**
 - Temporal layer 0 is 15 fps
 - Temporal layer 1 is 30 fps
 - Temporal layer 2 is 60 fps
- **1920x1080 resolution for all view layers**
- **With MVC style pixel rate level limit constraint, layers 1 and 2 have different levels**
 - Hence different PPS, SPS
- **With proposed pixel rate level limit constraint, layers 1 and 2 have same level**
 - Can use same PPS, SPS

Example

HEVC level constraints

Level	Pixel rate	picture size
1	552,960	36,864
2	3,686,400	122,880
3	13,762,560	458,752
3.1	33,177,600	983,040
4	62,668,800	2,088,960
4.1	62,668,800	2,088,960
4.2	133,693,440	2,228,224
4.3	133,693,440	2,228,224
5	267,386,880	8,912,896
5.1	267,386,880	8,912,896
5.2	534,773,760	8,912,896
6	1,002,700,800	33,423,360
6.1	2,005,401,600	33,423,360
6.2	4,010,803,200	33,423,360

layer _id	Individual Layer		Sub Bitstream		
	Pixel Rate	Level	temporal _id	Pixel Rate	Level
0	124,416,000	4.2	0	31,104,000	4
			1	62,208,000	4
			2	124,416,000	4.2
1	124,416,000	4.2	0	31,104,000	4
			1	62,208,000	4.2
			2	248,832,000	5
2	124,416,000	4.2	0	31,104,000	4.2
			1	62,208,000	5
			2	373,248,000	5.2

Conclusion

- **Recommend to adopt proposed SEI message for base HEVC and multiview extension for sub-bitstream profile and level indicators**
- **Recommend to define pixel rate level constraint in multiview extension such that constraint applies only to individual layer, and not to the sub-bitstream**

Backup

- If target temporal_id value = max_temporal_layers_minus1, for single layer, the extracted sub-bitstream is identical to the full bitstream
- No requirement that the profile and level for the sub-bitstream indicated in proposed SEI message be identical to the profile_idc and level_idc in the SPS
 - The bitstream must not violate either of the two constraints