Vidyo® Personal Telepresence

JCTVC-M0045: High-level syntax modifications for HEVC extensions

Jill Boyce, Danny Hong, Wonkap Jang

Proposals

- High level syntax design philosophy and/or syntax changes common to extensions to HEVC, including SHVC, MV-HEVC and 3D-HEVC

1. Allow different frame rate for different layers
2. Define an access unit as containing pictures from multiple layers, as in SVC and MVC
3. Conditionally include VPS extension syntax elements based on scalability_mask[i]
4. Add extension_type_flag[i] syntax elements in SPS extension and use for conditional syntax elements
5. Conditionally include syntax elements in slice segment header based on ViewId[nuh_layer_id], DepthFlag[nuh_layer_id], DependencyId[nuh_layer_id]
6. Profile constraints for Stereo main and proposed Scalable main profiles to disallow scalability combinations

1. Allow different frame rate for different layers

- Allow an enhancement layer to have a different frame rate than a reference layer
- Example:
  - 1280x720p30 base layer
  - 1920x1080p60 enhancement layer
- Intent in the current SHVC design is unclear on this point
- Proposed specification language included in proposal #2

2. Define an access unit as containing pictures from multiple layers, as in SVC and MVC

- Propose that an access unit be defined as containing coded pictures from multiple layers
- Ambiguity in MV-HEVC PDAM (JCT3V-C1004) about the appropriate definition of access unit
  - Does access unit contain a single picture from a single layer?
  - Does access unit contain pictures from multiple layers?
    - similar to MVC access unit definition
  - Editors note in MV-HEVC PDAM reflects uncertainty

    [Ed. (MH): The present version of this annex uses the same definition for an access unit as in clause 3, which essentially states that an access unit contains one coded picture (with a particular value of nuh_layer_id). One coded picture is defined below essentially identically to a view component in MVC. It is an open issue whether an access unit should instead be defined to contain all view components with the same POC value.]

2. Background: HEVC v1

- HEVC version 1 (JCTVC-L1003) scalable nesting SEI message semantics imply that an access unit may contain multiple layers, e.g. contain NAL units with different values of nuh_layer_id

all_layers_flag equal to 0 specifies that the list nestingLayerIdList[0] is specified by nesting_layer_id[i] for all i values in the range of 0 to nesting_num_layers_minus1, inclusive. all_layers_flag equal to 1 specifies that the list nestingLayerIdList[0] **consists of all values of nuh_layer_id present in the current access unit** that are greater than or equal to nuh_layer_id of the current SEI NAL unit, in increasing order of the values.

2. Background: MVC and MV-HEVC

- MVC definitions

access unit: A set of *NAL units* that are consecutive in *decoding order* and contain exactly one *primary coded picture* consisting of one or more *view components*. In addition to the *primary coded picture*, an access unit may also contain one or more *redundant coded pictures*, one *auxiliary coded picture*, or other *NAL units* not containing *slices* or *slice data partitions* of a *coded picture*. The decoding of an access unit always results in one *decoded picture* consisting of one or more *decoded view components*.

view component: A *coded representation* of a *view* in a single *access unit*.

- MV-HEVC includes the following definition of a coded picture, which differs from MVC.

coded picture: A *coded representation* of a *picture* comprising *VCL NAL units* with a particular value of nuh_layer_id and containing all *coding tree units* of the *picture*.

2. Proposed specification language

- For MV-HEVC and SHVC, propose definition such that an access unit be defined as containing coded pictures from multiple layers, and allowing different frame rates for the different layers
- Changes are marked with respect to HEVCv1 (JCTVC-L1003_v34).
  - **access unit**: A set of *NAL units* that are associated with each other according to a specified classification rule, are consecutive in *decoding order*, ~~and contain exactly one *coded picture*~~ contains zero or one *coded pictures* per *layer*, and contains a *coded picture* for at least one *layer*.

3. Conditionally include VPS extension syntax elements based on scalability_mask[i]

- Syntax elements directed only at some types of scalability and not others may be conditionally included in the VPS extension based upon which types of scalability are present, as indicated by scalability_mask[i]

- In particular, have layer set related syntax elements present for view scalability but not spatial/SNR scalability
  - In future, condition could be added for other scalability_mask[i] values

scalability_ mask index	Scalability dimension	ScalabilityId mapping
0	multiview	ViewId
1	spatial/SNR scalability	DependencyId
2-15	Reserved	

3. Proposed syntax and semantics

vps_extension( ) {	Descriptor
…	
if(scalability_mask[0]) {	
num_output_layer_sets	ue(v)
for(i = 0; i < num_output_layer_sets; i++) {	
output_layer_set_idx[i]	ue(v)
lsIdx = output_layer_set_idx[i]	
for(j = 0 ; j <= vps_max_layer_id; j++)	
if(layer_id_included_flag[lsIdx][j])	
output_layer_flag[lsIdx][j]	u(1)
}	
}	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = 0; j < i; j++)	
direct_dependency_flag[i][j]	u(1)
}	

num_output_layer_sets specifies the number of layer sets for which output layers are specified with output_layer_set_index[i] and output_layer_flag[lsIdx][j]. When not present, the value of num_output_layer_sets is inferred to be equal to 0.

When num_output_layer_sets not present, there is a single target output layer with nuh_layer_id equal to layer_id_in_nuh[vps_max_layers_minus1].

4. Add extension_type_flag[i] syntax elements in SPS extension and use for conditional syntax elements

- Propose to add several extension_type_flag[i] syntax elements in SPS
  - Conditionally include syntax elements related to specific extensions
- Avoid parsing dependency between SPS and VPS
- Allow future extensions to avoid parsing syntax elements for unrelated extensions
  - Traditional extension_flag design requires later extensions to parse all syntax elements associated with earlier extensions

4. Add extension_type_flag[i] syntax elements in SPS extension and use for conditional syntax elements

sps_extension( ) {	Descriptor
for(i = 0; i < 3; i++)	
extension_type_flag[i]	u(1)
if(extension_type_flag[0])	
inter_view_mv_vert_constraint_flag	u(1)
if(extension_type_flag[1]) {	
sps_inter_layer_mfm_enable_flag	u(1)
// additional syntax elements	
}	
if(extension_type_flag[2]) {	
// additional syntax elements	
}	
}	

extension_type_flag[0] equal to 1 indicates that the inter_view_mv_vert_constraint_flag syntax element is present. extension_type_flag[0] equal to 0 indicates that the inter_view_mv_vert_constraint_flag syntax element is not present.

extension_type_flag[1] equal to 1 indicates that the sps_inter_layer_mfm_enable_flag syntax element is present. extension_type_flag[1] equal to 0 indicates that the sps_inter_layer_mfm_enable_flag syntax element is not present.

extension_type_flag[2] shall be equal to 0 in bitstreams conforming to this version of this Specification. Other values for extension_type_flag[2] are reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore the value of extension_type_flag[2].

5. Conditionally include syntax elements in slice segment header

- Propose to add conditionally present syntax elements in the slice segment header of enhancement layers, based on values of
  - ViewId[i]
  - DepthFlag[i]
  - DependencyId[i]
- Allows future extensions to avoid parsing syntax elements for unrelated extensions
- Allows for future combined scalability profiles
- JCTVC-M0046 proposes particular syntax element only included for DependencyId > 0

5. Proposed syntax table

slice_segment_header( ) {	**Descriptor**
…	
if(nuh_layer_id > 0) {	
if(ViewId[nuh_layer_id] > 0) {	
// view scalable specific syntax elements	
}	
if(DepthFlag[nuh_layer_id] > 0) {	
// view scalable specific syntax elements	
}	
if(DependencyId[nuh_layer_id] > 0) {	
// spatial/SNR scalable specific syntax elements	
}	
}	
if(slice_segment_header_extension_present_flag) {	
…	

6. Profile constraints for Stereo main and proposed Scalable main profiles to disallow scalability combinations

- Existing constraints for the Stereo main profile in the MV-HEVC PDAM (JCT3V-C1004) does not disallow combined spatial/SNR and view scalability
- May wish to define combined scalability in the future, but is not the intent of current Stereo main profile
- Propose specific constraint language to reflect intent of Stereo main profile
- Propose similar constraint language for strawman Scalable main profile

6. Proposed additional constraint for stereo main profile

Stereo main profile

- Bitstreams conforming to the Stereo Main profile shall obey the following constraints:
  - The sub-bitstream resulting from the sub-bitstream extraction process with any value of tIdTarget and a value of 0 in layerIdListTarget as inputs shall conform to the Main profile.
  - The bitstream shall contain one layer with nuh_layer_id equal to i for which ViewId[i] is greater than 0.
  - When ViewId[i] is greater than 0, inter_view_mv_vert_constraint_flag shall be equal to 1 in the sps_extension( ) syntax structure of the active layer SPS of any coded pictures with nuh_layer_id equal to i.
  - When ViewId[i] is greater than 0, ScalabilityId[i][smIdx] shall be equal to 0 for smIdx in 1..15 for any coded pictures with nuh_layer_id equal to i.

6. Proposed constraint for strawman Scalable main profile

Scalable main profile

- Bitstreams conforming to the Scalable Main profile shall obey the following constraints:
  - The sub-bitstream resulting from the sub-bitstream extraction process with any value of tIdTarget and a value of 0 in layerIdListTarget as inputs shall conform to the Main profile.
  - When DependencyId[i] is greater than 0, ScalabilityId[i][smIdx] shall be equal to 0 for smIdx in 0..1 or 3..15 for any coded pictures with nuh_layer_id equal to i.

Summary of proposals

1. Allow different frame rate for different layers
2. Define an access unit as containing pictures from multiple layers, as in SVC and MVC
3. Conditionally include VPS extension syntax elements based on scalability_mask[i]
4. Add extension_type_flag[i] syntax elements in SPS extension and use for conditional syntax elements
5. Conditionally include syntax elements in slice segment header based on ViewId[nuh_layer_id], DepthFlag[nuh_layer_id], DependencyId[nuh_layer_id]
6. Profile constraints for Stereo main and proposed Scalable main profiles to disallow scalability combinations