

JVT-W035

Comments to MVC JD 2.0

Ying Chen, Ye-Kui Wang, and Miska M. Hannuksela

Outline

- Single-path adaptation
- On view_level
- IDR picture, IDR access unit and SPS
- Implicit removal
- Scalable nesting SEI in MVC

Single-path adaptation

- Priority based single-path adaptation is a fast and simple way to do the adaptation without parsing detailed information
 - View dependencies
 - Temporal dependencies
- Propose to introduce `priority_id` in the NAL unit header to support single-path adaptation similarly as in SVC
- Propose semantics constraints to both `priority_id` and `temporal_id`, similarly as in SVC
 - Let `pId/tId` be any possible value of `priority_id/temporal_id` in the coded video sequence. The bitstream that would be obtained by discarding all VCL NAL units with `priority_id/temporal_id` greater than `pId/tId` and the associated non-VCL NAL units and SEI messages shall be conforming to this Recommendation | International Standard.

On view_level

- The view_level (3 bits) in NALU header serves for two purposes
 - To identify whether a NALU with type 20 is a suffix NALU
 - view_level equal to 0 is thus reserved for a suffix NALU
 - To provide an indication of view dependency level
 - It is a subset of view dependency signaled in SPS MVC Ext.
 - Network elements that would perform intelligent adaptation based on information in NALU must be within the signaling context
 - The view_level can be deduced by view dependency at the network elements
- We propose to **replace** view_level with suffix_nal_unit_flag to serve for the first purpose

syntax changes

- Revisions to NAL unit header extension syntax

nal_unit_header_svc_mvc_extension() {	C	Descriptor
svc_mvc_flag	All	u(1)
if (!svc_mvc_flag) {		
...		
} else {		
priority_id	All	u(6)
temporal_id	All	u(3)
view_level	All	u(3)
suffix_nal_unit_flag	All	u(1)
anchor_pic_flag	All	u(1)
view_id	All	u(10)
idr_flag	All	u(1)
reserved_zero_one_bits	All	u(1)
}		
nalUnitHeaderBytes += 3		
}		

- Revisions to NAL unit header extension syntax

slice_layer_in_scalable_mvc_extension_rbsp() {	C	Descriptor
if (svc_mvc_flag == 1) {		
if (!suffix_nal_unit_flag) if (view_level != 0) {	2	
slice_header()		
slice_data()	2 3 4	
rbsp_slice_trailing_bits()	2	
}		
} else {		
...		

Proposal to JVT

JVT-V035.ppt / 2007-04-20 / ChenY

Semantics

suffix_nal_unit_flag equal to 1 indicates that the current NAL unit is a suffix NAL unit. **suffix_nal_unit_flag** equal to 0 indicates that the current NAL unit is not a suffix NAL unit.

IDR picture, IDR access unit and SPS

- V-IDR picture is defined in MVC, however, the following issues have not been decided and we propose to discuss them in JVT
 - When the AVC base view picture in an access unit is an IDR picture, can pictures of other views in the same access unit be non V-IDR pictures?
 - Shall an IDR access unit have all pictures in the access unit coded as IDR or V-IDR pictures?
 - When can an SPS change or be activated for a view that refers to the SPS? Does it only happen at an IDR access unit?
 - Shall the SPS MVC extension be identical in all the SPSs (in a coded video sequence) that contain an SPS MVC extension?
 - Shall the SPS MVC extension contain information of only the views that are required for decoding?

Implicit removal of decoded pictures

- Implicit removal
 - An inter-view only picture can be removed earlier: as soon as it is not used for further inter-view prediction
- The text for implicit removal of decoded non-reference pictures in the k non-output views in subclause H.8.3.2.3 is not clear enough to us.
 - Term “subsequent pictures in decoding order” is not clear to us
- Therefore, we propose a fix: subsequent pictures in decoding order consist of the following view_id values:
 - The view_id values specified in anchor_ref_LX [i][j] or non_anchor_ref_LX [i][j] (in SPS MVC extension)
 - The index i starts with the current view index (view decoding order of the view containing current decoding picture)
 - The values of i are further constrained by only considering the output or dependent views

H.8.3.2.3 “Removal of pictures from the DPB before possible insertion of the current picture”

...

Additionally, ~~all the non-reference pictures in the current access unit that belong to the k non-output views~~ ~~all inter-view only reference pictures m with the same $\text{PicOrderCnt}()$ as the decoded picture,~~ for which either of the following conditions are true, are removed from the DPB. Let ViewID be the view_id of the picture m and i be the index in the SPS MVC extension such that ViewID is equal to view_id[i].

- the decoded picture is an anchor picture and ~~the view id of picture m is not referenced by anchor_ref_1X[i] (with X being 0 or 1), where i denotes the view id of subsequent pictures in decoding order.~~ ViewID is not equal to any of anchor_ref_1X[k][j], wherein X is 0 or 1, j is any possible value allowed in the referred sequence parameter set, k is any possible value that is greater than i and that is allowed in the referred sequence parameter set and that the view with view_id equal to view_id[k] belongs to the p output views or k non-output views of the current operation point.
- the decoded picture is not an anchor picture and ~~the view id of picture m is not referenced by non_anchor_ref_1X[i] (with X being 0 or 1), where i denotes the view id of subsequent pictures in decoding order.~~ ViewID is not equal to any of non_anchor_ref_1X[k][j], wherein X is 0 or 1, j is any possible value allowed in the referred sequence parameter set, k is any possible value that is greater than i and that is allowed in the referred sequence parameter set and that the view with view_id equal to view_id[k] belongs to the p output views or k non-output views of the current operation point.

...

Scalable nesting SEI message

- The AVC SEI messages can be reused in a backward compatible manner and the scope of that message is specified by the scalable nesting SEI message
- It has similar syntax table as that of Scalable nesting SEI message in SVC

<code>scalable_nesting(payloadSize) {</code>	C	Descriptor
<code> all_pictures_in_au_flag</code>	5	u(1)
<code> if(all_pictures_in_au_flag == 0) {</code>		
<code> num_pictures_minus1</code>	5	ue(v)
<code> for(i = 0; i <= num_pictures_minus1; i++)</code>		
<code> pic_id[i]</code>	5	u(10)
<code> temporal_id</code>	5	u(3)
<code> }</code>		
<code> while(!byte_aligned())</code>		
<code> sei_nesting_zero_bit /* equal to 0 */</code>	5	f(1)
<code> sei_message()</code>	5	
<code> }</code>		

Scalable nesting SEI message

- The same syntax of this SEI message could be used for both SVC and MVC
- Mapping of the SVC syntax elements to the new MVC scalable nesting SEI message syntax

Syntax elements in the SVC spec	New syntax elements
dependency_id[i]	bits 0..2 of pic_id[i]
quality_id[i]	bits 3..4 of pic_id[i]
reserved_zero_five_bits	bits 5..9 of pic_id[i]

JVT-W035

Comments to MVC JD 2.0

Ying Chen, Ye-Kui Wang, and Miska M. Hannuksela

Thanks