

The background is a solid green color with a repeating pattern of white line-art icons. These icons include various nautical items like anchors, lifebuoys, seashells, and jellyfish, as well as technical symbols like Wi-Fi signals, lightning bolts, and a compass.

MEDIATEK

JVET-Q0171

AHG9: On CC-ALF modifications related to HLS

Olena Chubach, Ching-Yeh Chen, Chen-Yen Lai, Tzu-Der Chuang,
Yu-Wen Huang, Shaw-Min Lei

Presenter: Olena Chubach

Introduction

- Current CE5 common base specification description:
 - Filter coefficients of CC-ALF signalled in the adaptation parameter set (APS), and the appropriate APS is referenced either in PH or SH
 - Number of CC-ALF filters additionally signalled at PH and SH levels
 - CC-ALF parameters are signalled in PH no matter, whether chroma component is present or not, i.e. “ChromaArrayType!=0” is checked in SH, while this check is missing in PH

Proposed Aspect 1

- Remove following syntax elements from PH and SH:
pic_cross_component_cb_filters_signalled_minus1,
pic_cross_component_cr_filters_signalled_minus1,
slice_cross_component_cb_filters_signalled_minus1,
slice_cross_component_cr_filters_signalled_minus1
 - During the previous JVET meeting, it was claimed that slice level parsing dependency on parameter sets is natural, since it is unclear whether this would actually be used for error resilience in practice
 - Also, this will align CCALF syntax to chroma ALF

Proposed Aspect 2

- Add check “ChromaArrayType != 0” to PH, to condition signaling CC-ALF syntax in PH only when corresponding two chroma arrays are available
 - This way unnecessary signalling of CC-ALF parameters can be avoided

Related Syntax Modifications 1

Modifications related to aspect 1 are marked with green.

Modifications related to the aspect 2 are marked with yellow

picture_header_rbsp() {	Descriptor
.....	u(1)
if(sps_alf_enabled_flag) {	
pic_alf_enabled_present_flag	u(1)
if(pic_alf_enabled_present_flag) {	
pic_alf_enabled_flag	u(1)
if(pic_alf_enabled_flag) {	
pic_num_alf_aps_ids_luma	u(3)
for(i = 0; i < pic_num_alf_aps_ids_luma; i++)	
pic_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
pic_alf_chroma_idc	u(2)
if(pic_alf_chroma_idc)	
pic_alf_aps_id_chroma	u(3)
}	
if(ChromaArrayType != 0)	
pic_cross_component_alf_cb_enabled_flag	u(1)
if(pic_cross_component_alf_cb_enabled_flag) {	
pic_cross_component_alf_cb_aps_id	u(3)
pic_cross_component_cb_filters_signalled_minus1	ue(v)
}	
if(ChromaArrayType != 0)	
pic_cross_component_alf_cr_enabled_flag	u(1)
if(pic_cross_component_alf_cr_enabled_flag) {	
pic_cross_component_alf_cr_aps_id	u(3)
pic_cross_component_cr_filters_signalled_minus1	ue(v)
}	
}	
}	

Related Syntax Modifications 2

Modifications related to aspect 1 are marked with green.

Modifications related to the aspect 2 are marked with yellow

slice_header() {	Descriptor
...	u(v)
if(sps_alf_enabled_flag && !pic_alf_enabled_present_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
slice_num_alf_aps_ids_luma	u(3)
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)	
slice_alf_aps_id_luma[i]	u(3)
if(ChromaArrayType != 0)	
slice_alf_chroma_idc	u(2)
if(slice_alf_chroma_idc)	
slice_alf_aps_id_chroma	u(3)
}	
if(ChromaArrayType != 0)	
slice_cross_component_alf_cb_enabled_flag	u(1)
if(slice_cross_component_alf_cb_enabled_flag) {	
slice_cross_component_alf_cb_aps_id	u(3)
slice_cross_component_cb_filters_signalled_minus1	ue(v)
}	
if(ChromaArrayType != 0)	
slice_cross_component_alf_cr_enabled_flag	u(1)
if(slice_cross_component_alf_cr_enabled_flag) {	
slice_cross_component_alf_cr_aps_id	u(3)
slice_cross_component_cr_filters_signalled_minus1	ue(v)
}	
}	
...	
}	

Summary

Proposed two aspects for updating CC-ALF design:

- Remove following syntax elements from PH and SH:
pic_cross_component_cb_filters_signalled_minus1,
pic_cross_component_cr_filters_signalled_minus1,
slice_cross_component_cb_filters_signalled_minus1 and
slice_cross_component_cr_filters_signalled_minus1
- Add check “ChromaArrayType != 0” to PH, to condition signaling CC-ALF syntax in PH only when corresponding two chroma arrays are available
- It is suggested to adopt both aspects into the next version of VVC Specification draft