INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG04 MPEG VIDEO CODING

ISO/IEC JTC1/SC29/WG04 MPEG VC/M68232
July 2024, Sapporo, JP

Source	PUT, ETRI
Status	Input document
Title	[MIV] Signaling of extended geometry assistance
Authors	Jakub Stankowski, Błażej Szydełko, Adrian Dziembowski, Dawid Mieloch, Jun Young Jeong, Gwangsoon Lee

Abstract

The document proposes changes in *extended_geometry_assistance* syntax and semantics. The aims to allow for independent encoding and decoding of each *block_based_geometry_features* or other *geometry_features* if ever defined. The proponents recommend adopting this proposal.

1 Introduction

In WD4 [m66916] the "F.2.8.1 General extended geometry assistance SEI payload syntax" and "F.2.8.2 Block-based geometry features: initial block grid syntax" defines the following syntax:

F.2.8.1 General extended geometry assistance SEI payload syntax

extended_geometry_assistance(payloadSize) {	**Descriptor**
ega_num_views_minus1	ue(v)
ega_num_available_assistance_types_minus1	u(4)
for(v = 0; v <= ega_num_views_minus1; v++) {	
ega_assistance_present_flag[v]	u(1)
if(ega_assistance_present_flag[v]) {	
for(t = 0; t <= ega_num_available_assistance_types_minus1; t++) {	
ega_assistance_type_present_flag[v][t]	u(1)
}	
if(ega_assistance_type_present_flag[v][0]) {	
block_based_geometry_features(v)	
}	
}	
}	
}	

F.2.8.2 Block-based geometry features: initial block grid syntax

block_based_geometry_features(v) {	**Descriptor**
bbgf_qs[v]	ue(v)
bbgf_log2_bw_minus2[v]	ue(v)
bbgf_max_number_of_splits[v]	ue(v)
BbgfBW = 1 << (bbgf_log2_bw_minus2[v] + 2)	
bbgf_projection_plane_height_minus1[v]	ue(v)
bbgf_projection_plane_width_minus1[v]	ue(v)
for(l = 0; l < (bbgf_projection_plane_height_minus1[v] + BbgfBW) / BbgfBW; l++) {	
for(c = 0; c < (bbgf_projection_plane_width_minus1[v] + BbgfBW) / BbgfBW; c++) {	
recursive_split_function(v, l, c, 0)	
}	
}	
}	

In this approach the *block_based_geometry_features(v)* is interleaved with other syntax elements like ***ega_assistance_present_flag****[v]* or ***ega_assistance_type_present_flag****[v][t]*. This results in serve difficulties in parallel encoding of each *block_based_geometry_features(v)* object. Moreover, parallel (or selective) decoding of independent *block_based_geometry_features(v)* is not possible.

2 Proposed syntax changes

2.1 Step 1: Deinteleave syntax elements and introduce byte alignment for geometry_features

This modification allows to independently encode *block_based_geometry_features(v)* and combine individual byte-streams into final *extended_geometry_assistance(payloadSize)*.

F.2.8.1 General extended geometry assistance SEI payload syntax

extended_geometry_assistance(payloadSize) {	**Descriptor**
ega_num_views_minus1	ue(v)
ega_num_available_assistance_types_minus1	u(4)
for(v = 0; v <= ega_num_views_minus1; v++) {	
ega_assistance_present_flag[v]	u(1)
if(ega_assistance_present_flag[v]) {	
for(t = 0; t <= ega_num_available_assistance_types_minus1; t++) {	
ega_assistance_type_present_flag[v][t]	u(1)
}	
}	
}	
byte_alignment( )	
for(v = 0; v <= ega_num_views_minus1; v++) {	
if(ega_assistance_present_flag[v]) {	
if(ega_assistance_type_present_flag[v][0]) {	
block_based_geometry_features(v)	
}	
}	
}	
}	

F.2.8.2 Block-based geometry features: initial block grid syntax

block_based_geometry_features(v) {	**Descriptor**
bbgf_qs[v]	ue(v)
bbgf_log2_bw_minus2[v]	ue(v)
bbgf_max_number_of_splits[v]	ue(v)
BbgfBW = 1 << (bbgf_log2_bw_minus2[v] + 2)	
bbgf_projection_plane_height_minus1[v]	ue(v)
bbgf_projection_plane_width_minus1[v]	ue(v)
for(l = 0; l < (bbgf_projection_plane_height_minus1[v] + BbgfBW) / BbgfBW; l++) {	
for(c = 0; c < (bbgf_projection_plane_width_minus1[v] + BbgfBW) / BbgfBW; c++) {	
recursive_split_function(v, l, c, 0)	
}	
}	
byte_alignment( )	
}	

No changes to semantics are needed. The *byte_alignment( )* is defined in “Information technology — Coded representation of immersive media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)”

2.2 Step 2: Introduce ega_assistance_data_size syntax

This modification allows the decoder to separate independent *block_based_geometry_features(v)* units before parsing or decoding process. The main goal of this change is to make parallel or selective decoding possible.

F.2.8.1 General extended geometry assistance SEI payload syntax

extended_geometry_assistance(payloadSize) {	**Descriptor**
ega_num_views_minus1	ue(v)
ega_num_available_assistance_types_minus1	u(4)
ega_assistance_data_size_present_flag	u(1)
for(v = 0; v <= ega_num_views_minus1; v++) {	
ega_assistance_present_flag[v]	u(1)
if(ega_assistance_present_flag[v]) {	
for(t = 0; t <= ega_num_available_assistance_types_minus1; t++) {	
ega_assistance_type_present_flag[v][t]	u(1)
}	
}	
}	
if(ega_assistance_data_size_present_flag) {	
for(v = 0; v <= ega_num_views_minus1; v++) {	
if(ega_assistance_present_flag[v]) {	
for(t = 0; t <= ega_num_available_assistance_types_minus1; t++) {	
if(ega_assistance_present_flag[v][t]) {	
ega_assistance_data_size_bytes_minus1[v][t]	ue(v)
ega_assistance_data_size[v][t] = 0	
for(i = 0; i <= ega_assistance_data_size_bytes_minusi; t++) {	
ega_assistance_data_size_byte[v][t][i]	u(8)
ega_assistance_data_size[v][t] <<= 8	
ega_assistance_data_size[v][t] |= ega_assistance_data_size_byte[v][t][i]	
}	
}	
}	
}	
}	
}	
byte_alignment( )	
for(v = 0; v <= ega_num_views_minus1; v++) {	
if(ega_assistance_present_flag[v]) {	
if(ega_assistance_type_present_flag[v][0]) {	
block_based_geometry_features(v)	
}	
}	
}	
}	

F.3.9 Extended geometry assistance SEI payload semantics → F.3.9.1 General

ega_assistance_data_size_present_flag equal to 1 indicates that the information about size of each geometry_feature is present in the syntax structure. ega_assistance_data_size_present_flag equal to 0 indicates that the information about size of each geometry_feature is not present in the syntax structure.

ega_assistance_data_size_bytes_minus1[v][t] plus 1 specifies the number of bytes used to encode the size of geometry_features(v).

ega_assistance_data_size_byte[v][t][i] specifies the i-th byte (in big-endian order) of ega_assistance_data_size[v][t].

2.3 Step 3: Separate syntax into header and payload

extended_geometry_assistance(payloadSize) {	**Descriptor**
extended_geometry_assistance_header ( )	
byte_alignment( )	
extended_geometry_assistance_payload ( )	
}	

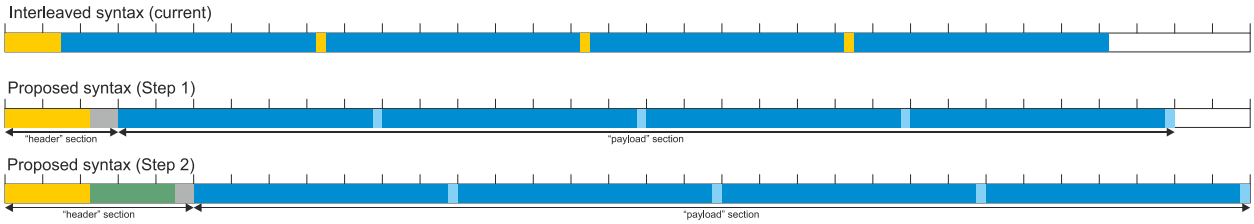
extended_geometry_assistance_header ( ) {	**Descriptor**
ega_num_views_minus1	ue(v)
ega_num_available_assistance_types_minus1	u(4)
ega_assistance_data_size_present_flag	u(1)
for(v = 0; v <= ega_num_views_minus1; v++) {	
ega_assistance_present_flag[v]	u(1)
if(ega_assistance_present_flag[v]) {	
for(t = 0; t <= ega_num_available_assistance_types_minus1; t++) {	
ega_assistance_type_present_flag[v][t]	u(1)
}	
}	
}	
if(ega_assistance_data_size_present_flag) {	
for(v = 0; v <= ega_num_views_minus1; v++) {	
if(ega_assistance_present_flag[v]) {	
for(t = 0; t <= ega_num_available_assistance_types_minus1; t++) {	
if(ega_assistance_present_flag[v][t]) {	
ega_assistance_data_size_bytes_minus1[v][t]	ue(v)
ega_assistance_data_size[v][t] = 0	
for(i = 0; i <= ega_assistance_data_size_bytes_minusi; t++) {	
ega_assistance_data_size_byte[v][t][i]	u(8)
ega_assistance_data_size[v][t] <<= 8	
ega_assistance_data_size[v][t] |= ega_assistance_data_size_byte[v][t][i]	
}	
}	
}	
}	
}	
}	
}	

extended_geometry_assistance_payload ( ) {	**Descriptor**
for(v = 0; v <= ega_num_views_minus1; v++) {	
if(ega_assistance_present_flag[v]) {	
if(ega_assistance_type_present_flag[v][0]) {	
block_based_geometry_features(v)	
}	
}	
}	

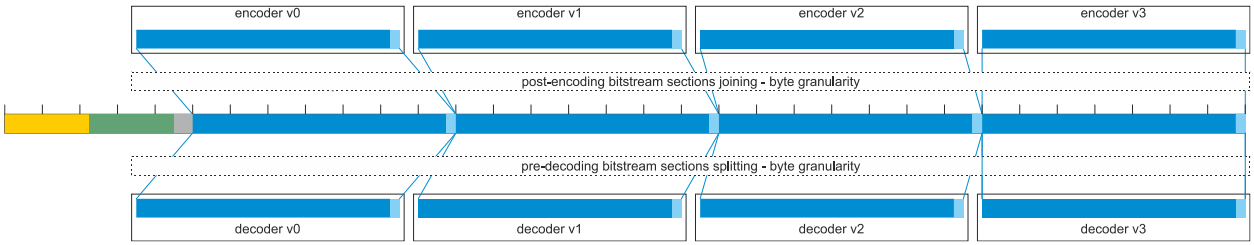
}	

3 Visualization(s)

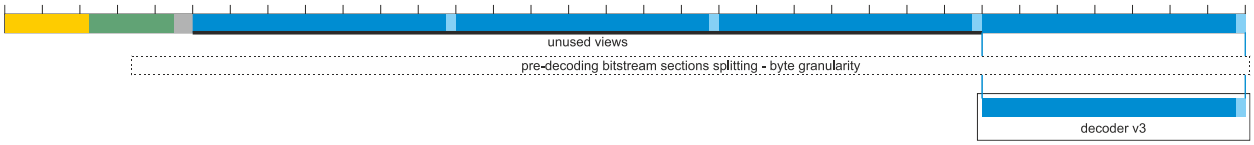
3.1 Proposed change



3.2 Use case – parallel encoding/decoding



3.3 Use case – selective decoding



4 Exemplary geometry assistance bitstream size

	Average size of single encoded view		Decoder throughput (Zmin + Zmax + skip)	
Sequence	1st frame (worst case)	Average	single view	all available views
A01	57825 B	3363 B	1200 MB/s	18 GB/s
J01	7335 B	504 B	296 MB/s	7.5 GB/s

5 Recommendation

The proponents recommend adopting this proposal.

6 Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory).