**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2018/m51005**

**October 2019, Geneva, CH**

|  |  |
| --- | --- |
| **Source** | Samsung Electronics, Nokia Technologies |
| **Status** | Input contribution |
| **Title** | Essential enhancement information messages |
| **Authors** | Rajan Joshi, Madhukar Budagavi (Samsung), Lukasz Kondrad (Nokia) |

# Introduction

The V-PCC specification defines two conformance points. The first conformance point (point A) covers the decoded attributes, geometry, and occupancy video streams, plus the decoded atlas information and decoded block to patch map, without any point cloud reconstruction. The second conformance point (point B) covers a fully reconstructed point cloud. This conformance point may include geometry and attribute smoothing, based on the values of certain syntax elements.

Although there is no consensus on this issue, the current V-PCC specification places geometry and attribute smoothing parameters in SEI messages. Since, this information is necessary for conformance to point B, a V-PCC decoder is required to decode such SEI messages. However, in practice, it is common for decoder implementations to discard the SEI messages.

# Proposal

To avoid a situation where SEI messages containing information that is required for conformance are discarded, it is proposed to create another kind of enhancement message, namely, essential enhancement message (EEI). Unlike SEI messages, EEI messages are an integral part of the V-PCC bitstream and shall be not removed from the bitstream. It is envisaged that there would be two types of essential enhancement messages:

1. Type-A EEI messages: These EEIs contain information required to check bitstream conformance and for output timing decoder conformance. Every conforming V-PCC decoder shall decode Type-A EEI messages.
2. Type-B EEI messages: V-PCC decoders using reconstruction profile Rec1 are required to decode Type-B EEI messages for conforming to point B.

As an example, the smoothing parameters SEI message proposed in [2] would instead be a Type-B EEI message.

# Specification Text

…

#### Essential enhancement information RBSP syntax

|  |  |
| --- | --- |
| eei\_rbsp( ) { | **Descriptor** |
| do |  |
| eei\_message( ) |  |
| while( more\_rbsp\_data( ) ) |  |
| rbsp\_trailing\_bits( ) |  |
| } |  |

#### 7.3.6.6 Supplemental enhancement information RBSP syntax

|  |  |
| --- | --- |
| sei\_rbsp( ) { | **Descriptor** |
| do |  |
| sei\_message( ) |  |
| while( more\_rbsp\_data( ) ) |  |
| rbsp\_trailing\_bits( ) |  |
| } |  |

…

### Essential enhancement information message syntax

|  |  |
| --- | --- |
| eei\_message( ) { | **Descriptor** |
| payloadType = 0 |  |
| do { |  |
| **eei\_payload\_type\_byte** | u(8) |
| payloadType += eei\_payload\_type\_byte |  |
| } while( eei\_payload\_type\_byte = = 0xFF ) |  |
| payloadSize = 0 |  |
| do{ |  |
| **eei\_payload\_size\_byte** | u(8) |
| payloadSize += eei\_payload\_size\_byte |  |
| } while( eei\_payload\_size\_byte = = 0xFF ) |  |
| eei\_payload( payloadType, payloadSize ) |  |
| } |  |

### Supplemental enhancement information message syntax

…

#### NAL unit header semantics

**nal\_forbidden\_zero\_bit** shall be equal to 0.

**nal\_unit\_type** specifies the type of RBSP data structure contained in the NAL unit as specified in Table 7-1.

NAL units that have nal\_unit\_type in the range of NAL\_UNSPEC\_48..NAL\_UNSPEC\_63, inclusive, for which semantics are not specified, shall not affect the decoding process specified in this Specification.

NOTE 1 – NAL unit types in the range of NAL\_UNSPEC\_48..NAL\_UNSPEC\_63 may be used as determined by the application. No decoding process for these values of nal\_unit\_type is specified in this Specification. Since different applications might use these NAL unit types for different purposes, particular care must be exercised in the design of encoders that generate NAL units with these nal\_unit\_type values, and in the design of decoders that interpret the content of NAL units with these nal\_unit\_type values. This Specification does not define any management for these values. These nal\_unit\_type values might only be suitable for use in contexts in which "collisions" of usage (i.e., different definitions of the meaning of the NAL unit content for the same nal\_unit\_type value) are unimportant, or not possible, or are managed – e.g., defined or managed in the controlling application or transport specification, or by controlling the environment in which bitstreams are distributed.

For purposes other than determining the amount of data in the decoding units of the bitstream (as specified in Annex C), decoders shall ignore (remove from the bitstream and discard) the contents of all NAL units that use reserved values of nal\_unit\_type.

NOTE 2 – This requirement allows future definition of compatible extensions to this Specification.

**Table 3‑1 – NAL unit type codes and NAL unit type classes**

|  |  |  |  |
| --- | --- | --- | --- |
| **nal\_unit\_type** | **Name of nal\_unit\_type** | **Content of NAL unit and RBSP syntax structure** | **NAL unit type class** |
| 0 | NAL\_TRAIL | Coded tile group of a non-TSA, non STSA trailing atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 1 | NAL\_TSA | Coded tile group of a TSA atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 2 | NAL\_STSA | Coded tile group of a STSA atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 3 | NAL\_RADL | Coded tile group of a RADL atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 4 | NAL\_RASL | Coded tile group of a RASL atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 5 | NAL\_SKIP | Coded tile group of a skipped atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 6..9 | NAL\_RSV\_ACL\_6.. NAL\_RSV\_ACL\_9 | Reserved non-IRAP ACL NAL unit types | ACL |
| 10 11 12 | NAL\_BLA\_W\_LP NAL\_BLA\_W\_RADL NAL\_BLA\_N\_LP | Coded tile group of a BLA atlas frame atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 13 14 15 | NAL\_GBLA\_W\_LP NAL\_GBLA\_W\_RADL NAL\_GBLA\_N\_LP | Coded tile group of a GBLA atlas frame atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 16 17 | NAL\_IDR\_W\_RADL NAL\_IDR\_N\_LP | Coded tile group of an IDR atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 18 19 | NAL\_GIDR\_W\_RADL NAL\_GIDR\_N\_LP | Coded tile group of a GIDR atlas frame  atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 20 | NAL\_CRA | Coded tile group of a CRA atlas frame atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 21 | NAL\_GCRA | Coded tile group of a GCRA atlas frame atlas\_tile\_group\_layer\_rbsp( ) | ACL |
| 22 23 | NAL\_IRAP\_ACL\_22 NAL\_IRAP\_ACL\_23 | Reserved IRAP ACL NAL unit types | ACL |
| 24..31 | NAL\_RSV\_ACL\_24.. NAL\_RSV\_ACL\_31 | Reserved non-IRAP ACL NAL unit types | ACL |
| 32 | NAL\_ASPS | Atlas sequence parameter set atlas\_sequence\_parameter\_set\_rbsp( ) | non-ACL |
| 33 | NAL\_AFPS | Atlas frame parameter set atlas\_frame\_parameter\_set\_rbsp( ) | non-ACL |
| 34 | NAL\_AUD | Access unit delimiter access\_unit\_delimiter\_rbsp( ) | non-ACL |
| 35 | NAL\_VPCC\_AUD | V-PCC access unit delimiter access\_unit\_delimiter\_rbsp( ) | non-ACL |
| 36 | NAL\_EOS | End of sequence end\_of\_seq\_rbsp( ) | non-ACL |
| 37 | NAL\_EOB | End of bitstream end\_of\_atlas\_substream\_rbsp( ) | non-ACL |
| 38 | NAL\_FD | Filler filler\_data\_rbsp( ) | non-ACL |
| 39 40 | NAL\_PREFIX\_SEI  NAL\_SUFFIX\_SEI | Supplemental enhancement information sei\_rbsp( ) | non-ACL |
| 41 42 | NAL\_PREFIX\_EEI NAL\_SUFFIX\_EEI | Essential enhancement information eei\_rbsp( ) | non-ACL |
| 43..47 | NAL\_RSV\_NACL\_43 NAL\_RSV\_NACL\_47 | Reserved non-ACL NAL unit types | non-ACL |
| 48..63 | NAL\_UNSPEC\_48.. NAL\_UNSPEC\_63 | Unspecified non-ACL NAL unit types | non-ACL |

NOTE 3 – A clean random access (CRA) and a global clean random access atlas frame may have associated random access skipped leading (RASL) or random access decodable leading (RADL) atlas frames present in the bitstream.

…

### Essential enhancement information message semantics

Each EEI message consists of the variables specifying the type payloadType and size payloadSize of the EEI message payload. EEI message payloads are specified in Annex E. The derived EEI message payload size payloadSize is specified in bytes and shall be equal to the number of bytes in the EEI message payload.

**eei\_payload\_type\_byte** is a byte of the payload type of an EEI message.

**eei\_payload\_size\_byte** is a byte of the payload size of an EEI message.

### Supplemental enhancement information message semantics

…

1. Essential enhancement information

(This annex forms an integral part of this Recommendation | International Standard.)

* 1. General

This annex specifies syntax and semantics for EEI message payloads.

The EEI messages are an essential part of the V-PCC bitstream. There are two types of essential enhancement messages:

1. Type-A EEI messages: These EEIs contain information required to check bitstream conformance and for output timing decoder conformance. Every conforming V-PCC decoder shall decode Type-A EEI messages.
2. Type-B EEI messages: V-PCC decoders using reconstruction profile Rec1 are required to decode Type-B EEI messages for conforming to point B.

Table XXX lists the type associated with each EEI message.

**Table 2: Classification of EEI messages**

|  |  |
| --- | --- |
| Type | EEI message |
| A |  |
| A |  |
| B | Smoothing parameters |

[Ed. (RJ): The syntax and semantics of the EEI messages should be here once the decision is made. The syntax and semantics (including persistence) will be identical to the corresponding SEI messages.]

# Reference

[1] “Text of ISO/IEC DIS 23090-5 Video-based Point Cloud Compression”, ISO/IEC JTC1/SC29/WG11 output document N18670, July 2019, Gothenburg, Sweden

[2] Lukasz Kondrad, Lauri Ilola, Kimmo Roimela, and Sebastian Schwarz, “Splitting SEI messages”, ISO/IEC JTC1/SC29/WG11 m50827, October 2019, Geneva, CH