 ISO/IEC JTC 1/SC 29/WG 3 N0458

**ISO/IEC JTC 1/SC 29/WG 3**

**MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output Document

**Title:** Exploration Experiments for MPEG-I Scene Description

**Status:** Approved

**Date of document:** 2022-01-28

**Source:** ISO/IEC JTC 1/SC 29/WG 3

**No. of pages:** 32 (with cover page)

**Email of Convenor:** young.L@samsung.com

**Committee URL:** <https://isotc.iso.org/livelink/livelink/open/jtc1sc29wg3>

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 3**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 3 N** **0458**

**January 2022, Virtual**

|  |  |
| --- | --- |
| **Title** | **Exploration Experiments for MPEG-I Scene Description** |
| **Source** | **WG 03, MPEG Systems** |
| **Status** | **Approved** |
| **Serial Number** | **21174** |

[1 EE1: Carriage of Random Access Support in Scene Description (closed) 2](#_Toc94260852)

[2 EE2: Dynamic Scene Update (closed) 2](#_Toc94260853)

[3 EE3: Codec Support in MPEG-I SD (ongoing) 2](#_Toc94260854)

[4 EE4: Haptics Support (new) 12](#_Toc94260855)

[5 EE5: Generic Interactivity Framework (new) 15](#_Toc94260858)

[6 EE6: User Representation and Avatars (new) 26](#_Toc94260859)

[7 EE7: AR Anchoring (new) 29](#_Toc94260860)

1. **EE1: Carriage of Random Access Support in Scene Description (closed)**

EE closed at MPEG #137. Please see WG03 N00383 for the latest description of this EE.

1. **EE2: Dynamic Scene Update (closed)**

EE closed at MPEG #137. Please see WG03 N00383 for the latest description of this EE.

1. **EE3: Codec Support in MPEG-I SD (ongoing)**
   1. **Background**

At the 136th MPEG meeting, WG7, WG4, and WG3 agreed jointly to establish a new EE as part of the MPEG-I Scene Description AHG to study and specify the necessary extensions to add support for the V3C codecs (V-PCC and MIV) in particular, and all immersive MPEG codecs in general.

The background of the discussion can be found in [1].

* 1. **Current Understanding**

It was established that when adding codec support, the following options are possible:

* Codec independent: in this option, all the necessary decoding and post-processing is performed to produce a primitive format that is natively supported by the Presentation Engine.
* Codec dependent: in this option, the Presentation Engine needs to have some level of support for the codec, in order to be able to render the object.
  + Variant a: in this variant, an intermediate uncompressed format is passed to the Presentation Engine for rendering. A Presentation Engine that supports this format may then load the appropriate shader programs to perform post-processing (e.g. 3D reconstruction) and rendering of the object. The Presentation Engine must support the intermediate format.
  + Variant b: in this variant, samples of the compressed stream are passed to the Presentation Engine for decompression, post-processing, and rendering. The Presentation Engine must support the compressed format.

These variants vary in the split of tasks between the media pipeline in the MAF and the Presentation Engine. The following diagram depicts example pipelines for these different options.



**Figure 1 Example Pipeline Options**

The following example show how the different options could be described in the MPEG-I SD glTF document:

**Table 1 Example glTF for Option 1**

|  |
| --- |
| *.*  *.*  *.*  {  *"name"*: ”vpcc\_longdress",  *"primitives"*: [  {  *"attributes"*: {  *"POSITION"*: 15,  *"COLOR\_0"*: 16  },  *"mode"*: 0  }  ]  }  *.*  *.*  *.*  *"extensions"*: {  *"MPEG\_media"*: {  *"media"*: [  {  *"name"*: "longdress",  *"timeOffset"*: 0.0,  *"alternatives"*: [  {  *"mimeType"*: "video/mp4;codec=v3e1.L2.0.0.1, avc1.4D401E",  *"uri"*: "https://example.com/vpcc\_longdress.mp4"  }  ]  }  ]  }  }, |

**Table 2 Example glTF for Option 2a**

|  |
| --- |
| {  *"name"*: "vpcc\_longdress",  *"primitives"*: [  {  *"attributes"*: {  *"\_MPEG\_V3C\_ATLAS\_0"*: 1136,  *"\_MPEG\_V3C\_GEOMETRY\_0"*: 1134,  *"\_MPEG\_V3C\_OCCUPANCY\_0"*: 1135,  *"\_MPEG\_V3C\_COLOR\_0"*: 1137  },  *"mode"*: 0  }  ]  }  *.*  *.*  *.*  *"extensions"*: {  *"MPEG\_media"*: {  *"media"*: [  {  *"alternatives"*: [  {  *"mimeType"*: "video/mp4;codec=v3e1.L2.0.0.1, avc1.4D401E",  *"tracks"*: [  {  *"track"*: "#track\_ID=1"  }  ],  *"uri"*: "https://example.com/vpcc\_longdress.mp4"  }  ],  *"loop"*: true,  *"timeOffset"*: 0  }  ]  }  }, |

**Table 3 Example glTF for Option 2b**

|  |
| --- |
| {  *"name"*: "vpcc\_longdress",  *"primitives"*: [  {  *"attributes"*: {  *"\_MPEG\_V3C\_POINTCLOUD"*: 165,  },  *"mode"*: 0  }  ]  }  *.*  *.*  *.*  *"extensions"*: {  *"MPEG\_media"*: {  *"media"*: [  {  *"alternatives"*: [  {  *"mimeType"*: "video/mp4;codec=v3e1.L2.0.0.1, avc1.4D401E ",  *"tracks"*: [  {  *"track"*: "#track\_ID=1"  }  ],  *"uri"*: " https://example.com/vpcc\_longdress.mp4"  }  ],  *"loop"*: true,  *"timeOffset"*: 0  }  ]  } |

* 1. **Mandates**

This EE will have the following mandates:

* Identify the MPEG codecs to be supported in MPEG-I SD
* Define the MIME type and any necessary signaling and extensions to enable options 1 and 2
* For codec dependent support:
  + Evaluate the codec-dependent options and decide on which ones to enable
  + Define the exact buffer formats and any necessary restrictions on the formats
  + Define any necessary glTF extensions and register any new \_MPEG attributes
  + Provide guidelines on how to implement the variant
* Provide test scenarios, assets, and implementation in the reference software
  1. **Support for V-PCC**
  2. **Support for V3C**
     1. **m58918 - InterDigital Response to EE3**

An option for media pipeline is envisioned where the MAF performs the decoding, and any necessary processing and the presentation engine performs the 3D reconstruction. The option is informally called as pipeline option #2A.

In the V3C specification, V3C profiles follow a structured and flexible definition to allow for clearly identifying two distinct conformance points namely conformance point A and conformance point B (see Annex A in ISO/IEC 23090-5 [2]). The first conformance point, point A, covers the decoded video sub-bitstreams and atlas sub-bitstream. It also covers the derived block to patch map information in atlas sub-bitstream. The second conformance point, point B, covers the reconstruction process.

Following the definition in V3C and the design goals for pipeline option #2A, the MAF can be assumed to perform operations associated with conformance point A in the V3C specification. On the other hand, the Presentation Engine is responsible for performing the operations associated with conformance point B, as shown in Figure 1.

It is necessary to express the intermediatory formats for the different V3C components such that the Presentation Engine can use the information in relevant buffer/texture formats for 3D reconstruction. The MAF performs decoding and processes the decoded V3C components. The result of MAF processing is a representation of the different decoded and processed V3C components in formats that are consumable by the Presentation Engine.

In response to EE3, this contribution proposes a new extension to support V3C content in ISO/IEC 23090-14.

Diagram

Description automatically generated

Figure 2 An overview of the glTF document structure with MPEG extensions and MPEG\_V3C extensions

**MPEG\_V3C scene-level extension**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Usage** | **Description** |
| objects | Array | [] | - | Array of V3C objects |

**MPEG\_V3C node-level extension**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Usage** | **Description** |
| object | number | - | M | An index of a V3C object in the objects array in the scene-level MPEG\_V3C extension. |

**V3C\_ATLAS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Usage** | **Description** |
| patchBlockSize | Number | 16 | M | Describes the patch block size of the atlas frame |
| blockToPatchInformation | Number | - | M | Index in the accessor array which refers to the block to patch information data |
| totalPatches | Number | - | M | Index in the accessor array which holds the information on number of patches |
| commonPatchParameters | Number | - | M | Index in the accessor array which holds the information on common patch parameters |
| patchInformation | Array | [] | M | Array of patch types and their respective information |

**V3C\_ATLAS.patchInformation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Usage** | **Description** |
| type | Number | - | M | Describes the type of patch |
| PLRDLevel | Number | - | O | Index in the accessors array which holds the information whether the PLR is at block level or patch level.  Only applicable if patch type is PROJECTED. |
| PLRDPresentBlockFlag | Number | - | O | Index in the accessors array which holds the information on presence of block level PLR mode.  Only applicable if patch type is PROJECTED. |
| PLRDBlockModeMinus1 | Number | - | O | Index in the accessors array which holds the information on block level PLR mode.  Only applicable if patch type is PROJECTED. |
| PLRDPresentFlag | Number | - | O | Index in the accessors array which holds the information on presence of patch level PLR Mode.  Only applicable if patch type is PROJECTED. |
| PLRDModeMinus1 | Number | - | O | Index in the accessors array which holds the information on patch level PLR mode.  Only applicable if patch type is PROJECTED. |
| patchAssociatedPatchIndex | Number | - | O | Index in the accessors array which specifies the index of the patches associated EOM patches.  Only applicable if patch type is EOM. |
| patchEOMPoints | Number | - | O | Index in the accessors array which holds the information on the number of EOM coded points.  Only applicable if patch type is EOM. |
| numberRAWPoints | Number | - | O | Index in the accessors array which holds the information on the number of raw coded points.  Only applicable if patch type is RAW. |

**V3C\_OCCUPANCY and V3C\_GEOMETRY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Usage** | **Description** |
| index | Number | - | M | The index of a texture object in the textures array that is associated with the V3C component |

**V3C\_ATTRIBUTE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Usage** | **Description** |
| index | Number | - | M | The index of a texture object in the textures array that is associated with the V3C attribute component |
| type | Number | - | M | Key identifier for the V3C attribute type, as defined in Table 3 of ISO/IEC 23090-5. |

**V3C\_ATTRIBUTE.type**

|  |  |
| --- | --- |
| **Attribute values** | **Attribute type** |
| 0 | Texture |
| 1 | Material ID |
| 2 | Transparency |
| 3 | Reflectance |
| 4 | Normals |
| 5..14 | Reserved |
| 15 | Unspecified |

* 1. **Support for MIV**
     1. **MIV player from m58430**

As part of the scene description EE on codec support[[1]](#footnote-1), Philips has provided a software example for a MIV renderer.

The provided real-time renderer is a simple example to "get going" and does not have sufficient quality in general, but as this EE is more of an implementation effort, we believe that this example is suitable.

The OpenGL ES 3.2 shaders are included as raw string literals within vr.scene.zmin.cpp, and the C++ code is only provided to create a running example.

* + 1. **Build instructions**

1. Unpack the archive at any location.
2. Add pre-built or self-built external libraries:
   * GLEW 2.1 to C:\X\ext\glew
   * GLFW 3.3 or newer to C:\X\ext\glfw
   * OpenCV 3.4.13 to C:\X\ext\opencv-3.4.13\_install
   * In case of confusion, please study the file C:\X\files.txt. It has the output of Get-ChildItem -Recurse. We prefer not to provide binaries to avoid software licensing problems.
3. For the purpose of this description, this document is located at C:\X\README.md
4. Open the folder C:\X\ee\_on\_miv\_support\_in\_sd in a terminal
5. mkdir build
6. cd build
7. cmake -DOpenCV\_DIR=C:\X\ext\opencv-3.4.13\_install ..
8. Open Visual Studio 2019 to build in Release mode with platform x64
9. Copy C:\X\ext\glew\bin\win64\glew32.dll to C:\X\ee\_on\_miv\_support\_in\_sd\build\AppsGL\Release\
10. Copy "C:\X\ext\opencv-3.4.13\_install\x64\vc16\bin\opencv\_world3413.dll" to C:\X\ee\_on\_miv\_support\_in\_sd\build\AppsGL\Release\
    * 1. **Run instructions**
11. On a multi-GPU system, make sure that the right GPU is selected. For NVIDIA:
    1. Control panel
    2. 3D Settings
    3. Manage 3D Settings
    4. Preferred graphics processor
    5. High-Performance NVIDIA processor
12. Start run.bat
    * 1. **Location**

* Software: <http://mpegx.int-evry.fr/software/MPEG/MIV/other/miv-player-example>
* Documentation: <http://mpegx.int-evry.fr/software/MPEG/MIV/other/miv-player-example/README.md>
* Zip-file with software and example data on the MPEG content server:  
  /MPEG-I/Part12-ImmersiveVideo/test\_material/m58999 SD-EE on Codec Support
  + 1. **Test data**

The example data includes one frame of Museum (ERP, 3DoF+) and one of Kitchen (PSP, 6DoF window) each with three pose traces.

* + 1. **License**

The software has the typical ISO/IEC modified BSD license.

External libraries are not included to avoid software license issues, but the documentation includes build instructions.

* 1. **Support for G-PCC**
  2. **Participants**

The following EE participants are identified:

|  |  |  |
| --- | --- | --- |
| **Participant** | **Affiliation** | **NB** |
| Imed Bouazizi | Qualcomm | US |
| Basel Salahieh | Vimmerse | US |
| Lauri Illola | Nokia | DE |
| Lukasz Kondrad | Nokia | DE |
| Ahmed Hamza | Interdigital | CA |
| Gurdeep Bhullar | Interdigital | CA |
| Bart Kroon | Philips | NL |

* 1. **Timeline**
* 2022-01-12: MPEG document upload deadline
* 2022-01-17: MPEG #137(online) meeting starts
  1. **References**

[1] m58329, Codec Support in SD

[2] MIV Test Model, <https://dms.mpeg.expert/doc_end_user/documents/135_OnLine/wg11/MDS20596_WG04_N00112.zip>

1. **EE4: Haptics Support** **(new)**
   1. **Introduction**

In order to incorporate haptics as part of the immersive media experience, it is necessary to enable the content creator to associate haptic media with objects in the scene. In m58487 [1], four extensions to glTF that enabled haptic interactions in immersive media scenarios were proposed. This EE focuses on the two extensions, MPEG\_haptic, and MPEG\_material\_haptic that are specific to haptics support in MPEG-I SD. The other two extensions, MPEG\_interaction, and MPEG\_avatar, are the subjects of other EEs (see Section 5 below).

1. 2. **Problem statement**

Visual properties of objects can be described in glTF. Some of those properties, such as geometry and size, can be used for haptic rendering. But more information is required to enable rich haptic feedback. Friction, roughness, stiffness, temperature are just a few examples of such properties. The exhaustive list of such haptics-related object properties and associated metrics should still be defined.

glTF needs to be extended with this haptic information. The proposed extension MPEG\_mesh\_collision [2] mentions a material property that could be dedicated to haptic data. This concept needs to be developed in a dedicated extension.

* 1. **Use cases relevant for the EE**

Both the Haptics Phase 2A and Phase 2B use cases, documented in detail in WG 02 N00139 [3] are relevant for this EE.

* 1. **Related (WG2) and Extracted (new) Requirements**

All the requirements in Section 4.8 of WG02 N00130 [4] that are tagged as Phase 2, Phase 2A, or Phase 2B are relevant to this EE.

* 1. **Relation to other activities (EE, requirements, etc…)**
* [EE on User Representation and Avatars](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/MPEG-Contributions/-/issues/260)
* [EE on Interactivity](http://mpegx.int-evry.fr/software/MPEG/Systems/SceneDescription/MPEG-Contributions/-/issues/261)
  1. **Mandates**

The mandates for this EE are as follows:

* Define an exhaustive list of haptics-related object properties and associated metrics.
* Flesh out the MPEG\_haptic extension introduced in [1].
* Flesh out the MPEG\_material\_haptic introduced in [1].
* Evaluate the two extensions for completeness and functionality.
* Define test scenarios and collect test assets.
* Develop the reference software integration and validate against the test scenarios.
* Ensure compatibility with the other related EEs that involve haptics.
  1. **Participants**

|  |  |  |  |
| --- | --- | --- | --- |
| Participant | Contact | Email | Type |
|  |  |  |  |
| Immersion Corporation | Yeshwant Muthusamy | [ymuthusamy@immersion.com](mailto:ymuthusamy@immersion.com) | L |
| InterDigital | Fabien Danieau | [Fabien.danieau@interdigital.com](mailto:Fabien.danieau@interdigital.com) | P |
| InterDigital | Patrice Hirtzlin | [Patrice.hirtzlin@interdigital.com](mailto:Patrice.hirtzlin@interdigital.com) | P |
| Qualcomm, Inc. | Imed Bouazizi | [bouazizi@qti.qualcomm.com](mailto:bouazizi@qti.qualcomm.com) | P |
| Immersion Corporation | Chris Ullrich | [cullrich@immersion.com](mailto:cullrich@immersion.com) | P |

(P = proponent, L = leader)

* 1. **Information about proposed technologies**

List of already submitted contributions on this topic.

* + 1. **m58487 – MPEG-I SD Haptics Schema and Processing Model**

In this contribution, we make the case for adding haptics to glTF objects in a scene, in order to complete the immersive media experience. Four new extensions, MPEG\_haptic, MPEG\_material\_haptic, MPEG\_avatar, and MPEG\_interaction are introduced and briefly described. Initial versions of the associated semantics and JSON schemas for the first three extensions are also presented. Finally, initial version of the processing model, describing how these four extensions would interact with each other, is also described. The key diagram from this contribution is shown below:

Diagram

Description automatically generated

**Figure 1: Proposed location of the extensions for haptics, interaction, and avatars, among all MPEG extensions to glTF.**

EE participants are encouraged to review m58487 [1] for the semantics tables, JSON schemas, and the processing model.

* 1. **Test cases**

The EE may define test cases for which the evaluation criteria will be analyzed. For instance, a first test case can be with live content while another in the on-demand content.

* 1. **Evaluation criteria**

List of criteria that will allow to compare the different technical solutions and converge to a unique solution. Criteria can be objective like memory efficiency, bitrate or subjective flexibility, compatibility with legacy solution, etc.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Description** | **Evaluation** |
| Criteria #1 | Description | The technical solution should minimize/optimise … |

* 1. **Timeline**
* **2022-04-25: MPEG #138**
* Define an exhaustive list of haptics-related object properties
* Flesh out the MPEG\_haptic extension
* Flesh out the MPEG\_material\_haptic extension.
* Evaluate the two extensions for completeness and functionality.
* Define test scenarios and collect test assets.
* **2022-07-18: MPEG #139**
* Develop the reference software integration and validate against the test scenarios.
* Ensure compatibility with the other related EEs that involve haptics.
  1. **References**

1. m58487, “MPEG-I SD Haptic Schema and Processing Model”, MPEG137, January 2022.
2. m56337. MPEG-I WG 03 MPEG Systems. Interactivity Support in Scene Description. April 2021.
3. WG 02 N00139, “Updated MPEG-I Phase 2 Haptics Use Cases”, MPEG136, October 2021.
4. WG 02 N00130, “MPEG-I Phase 2 Requirements”, MPEG136, October 2021.
5. **EE5: Generic Interactivity Framework (new)** 
   1. **Introduction**

The MPEG Scene Description solution has added support for timed media to glTF 2.0. A Media Access Function (MAF) offers an API to the Presentation Engine, through which timed media can be requested. The current scene description solution allows the user to consume the scene in 6DoF, thus, moving freely in the 3D scene. To offer a realistic experience, the viewer should be able to interact with objects in the scene in different ways.

* 1. **Problem statement**

The following aspects of interactivity are identified:

* the user cannot walk through obstacles in the scene (such as walls, chairs, tables, …)
* the user is able to interact with objects in the scene in a way that results in changes to the scene (e.g. turn on a TV, open a door, push objects, …)
* the user will perceive the changes caused by the interaction (e.g. visual, audio, and haptics feedback)

This EE will focus on developing the necessary extensions to support basic interactivity in scene description.

The following simplified architecture is identified as the baseline for a generic interactivity framework.

Shape

Description automatically generated with medium confidence

Triggers are events that will trigger some form of interactivity. Actions are the interactivity feedback. The TuC [1] currently contains a collision model that defines one form of interactivity trigger. Objects provide a simplified mesh that will allow for detection of collision with the viewer. A detected collision will trigger some interactivity actions, such as starting an animation, haptics and/or audio feedback.

* 1. **Use cases relevant for the EE**
  2. **Related (WG2) and Extracted (new) Requirements**

The following requirements are relevant and addressed by this EE:

* It shall be possible to discover user interactivity modules (requirement #85)
* It shall be possible to define custom interactivity procedures based on input from the user or from the user’s devices and sensors (requirement #86)
* Support of user interactivity with objects within a virtual environment (requirement #90)
* Support of interaction between multiple users within an immersive environment (requirement #95)
* The specification shall support interactivity models related to avatar position and orientation (requirement #129)
* The specification shall support coding and presentation of interactivity models related to avatar-scene or avatar-avatar interactions (requirement #130)
* The specification shall support different media types and various haptic feedback paradigms (requirement #131)
  1. **Relation to other activities (EE, requirements, etc…)**

A relationship to the Haptics phase 2 activity has been identified.

* 1. **Mandates**

The mandates for this EE are as follows:

* refine the generic interactivity framework reference architecture
* define a basic set of interactivity triggers
* define a basic set of interactivity actions that covers different media types
* define test scenarios and collect test assets
* evaluate proposed solutions
* develop the reference software integration and validate against the test scenarios
  1. **Participants**

|  |  |  |  |
| --- | --- | --- | --- |
| Participant | Contact | Email | Type |
|  |  |  |  |
| Qualcomm | Imed Bouazizi | bouazizi@qti.qualcomm.com | L |
| Immerse | Yeshwant Muthusamy | ymuthusamy@immersion.com | P |
| Interdigital | Fabien Danieau | fabien.danieau@interdigital.com | P |
| Interdigital | Patrice Hirtzlin | patrice.hirtzlin@interdigital.com | P |
| Interdigital | Gurdeep Bhullar | gurdeep.bhullar@interdigital.com | P |
| Xiaomi | Emmanuel Thomas | thomase@xiaomi.com | P |

(P = proponent, L = leader)

* 1. **Information about proposed technologies**

The following contributions on Interactivity have been submitted:

**Meeting #134**

[m56337](https://dms.mpeg.expert/doc_end_user/current_document.php?id=78240&id_meeting=0) [SD] Interactivity in Scene Description

**Meeting #135**[m57409 [SD] Interactivity support in scene description](https://dms.mpeg.expert/doc_end_user/current_document.php?id=79601&id_meeting=187)

**Meeting #136**

[m58104](https://dms.mpeg.expert/doc_end_user/current_document.php?id=80564&id_meeting=0) [SD] On scene interactivity

[m58146](https://dms.mpeg.expert/doc_end_user/current_document.php?id=80606&id_meeting=0) [SD] Describing camera paths for interactivity

**Meeting #137**

[m58486 [SD] Collision model for Interactivity](https://dms.mpeg.expert/doc_end_user/current_document.php?id=81216&id_meeting=189)

[m58794](https://dms.mpeg.expert/doc_end_user/current_document.php?id=81524&id_meeting=0) [SD] On interactivity support

* 1. **Extracted from TuC**
     1. **General**

In order to provide an immersive experience to the viewer, it is important that the viewer interacts properly with objects in the scene. The viewer should not be able to walk through solid objects in the scene, such as walls, chairs, and tables.

The following figure depicts a 3D mesh representation of a chair, together with its collision boundaries, defined as a set of cuboids.

Graphical user interface

Description automatically generated

* + 1. **Semantics**

The “MPEG\_mesh\_collision” extension is defined to provide a description of the collision boundaries of a mesh. The extension shall be defined on mesh objects as a set of cuboids around the mesh geometry.

It contains the following properties.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Default** | **Description** |
| boundaries | Array(object) | N/A | Array of boundary shapes that are used to define the collision boundaries of the mesh object. The boundaries may be spheroids or cuboids, as defined in the MPEG\_camera\_control extension. |
| static | boolean | True | Determines if the object is affected by collisions or not. An object that is static will not be affected by collisions, which means that when the viewer or another object collides with this object, its position will not be altered. |
| material | number | N/A | The index of a collision material that defines how colliding objects or viewers will interact with this object. This may include bounciness, friction, etc. |
| animations | Array(object) | N/A | Defines animations that are triggered by a collision or action on this object. The animation may be limited to a subset of other objects, e.g. only the viewer may trigger this animation. It also contains a pointer to the animation that is to be executed when triggered. |
|  |  |  |  |

The mesh collision information consists of the cuboid vertex coordinates (x,y,z) for cuboid boundaries or the sphere center and radius for spherical boundaries. The values are provided as float numbers.

* + 1. **Processing Model**

The Presentation Engine shall support the MPEG\_mesh\_collision extension. The camera position (x,y,z) shall not be contained within one of the defined mesh cuboids at any point of time. Collision may be signaled to the viewer through visual, acoustic, and/or haptic feedback.

This information on the boundaries for the nodes may be used to initialize and configure a 3D physics engine that will detect collisions.

* 1. **Contribution m58794**
     1. **MPEG scene interactivity glTF extension**

A MPEG interactivity glTF extension, called MPEG\_scene\_interactivity, is introduced at the scene level as shown in Figure 5.



Figure 5 : MPEG interactivity glTF extension at scene level

MPEG\_scene\_interactivity glTF scene-level extension adopts a semantic approach based on the definition of behaviors, triggers and actions.

* + 1. **MPEG\_scene\_interactivity definition**

A behavior defines which kind of interactivity is allowed at runtime for dedicated virtual objects, corresponding to glTF nodes.

A behavior corresponds to a unique association of triggers and actions:

* the triggers define the runtime conditions to be met before executing the behavior actions
* the actions define how the behavior affects the scene
  + 1. **MPEG\_scene\_interactivity semantic**

The semantic of the MPEG\_scene\_interactivity glTF extension is shown in Table 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| triggers | Array | M | [] | Contains the definition of all the triggers used in that scene |
| actions | Array | M | [] | Contains the definition of all the actions used in that scene |
| behaviors | Array | M | [] | Contains the definition of all the behaviors used in that scene. A behavior is composed of a pair of (triggers, actions), control parameters of triggers and actions, a priority weight and an optional interrupt action as detailed in 3.3 |

Table 3 : Semantic of the MPEG\_scene\_interactivity extension

* + 1. **Trigger semantic**

The semantic of a trigger is provided in Table 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M | VISIBILITY | Defines the type of the trigger by taking one of the following values:  VISIBILITY = 0,  PROXIMITY = 1,  USER\_INPUT = 2,  TIMED = 3,  COLLIDER = 4 |
| activateOnce | Boolean | M | FALSE | If FALSE: the trigger is activated each time its conditions are met.  If TRUE: the trigger is activated once when its conditions are met.  Refer to Figure 6 |
| If(type== VISIBILITY){ |  |  |  |  |
| cameraNode | Number | M |  | Index to the node containing a camera in the nodes array for which the visibilities are determined |
| nodes | Array | M |  | Indices of the nodes in the nodes array to be considered. All the nodes shall be visible by the camera to activate the trigger |
| } |  |  |  |  |
| If(type == PROXIMITY){ |  |  |  |  |
| distanceLowerLimit | Number | M | 0 | Threshold min in meters for the node proximity calculation |
| distanceUpperLimit | Number | O |  | Threshold max in meters for the node proximity calculation |
| nodes | Array | M | [] | Indices of the nodes in the nodes array to be considered. All the nodes shall have a distance from the user camera above the distanceLowerLimit and below the distanceUpperLimit to activate the trigger |
| } |  |  |  |  |
| If(type == USER\_INPUT){ |  |  |  |  |
| userInputDescription | String | M |  | Describe the user body part and gesture related to the input. E.g. “/user/hand/left/grip” |
| nodes | Array | O |  | Indices of the nodes in the nodes array to be considered for this user input |
| } |  |  |  |  |
| If(type == TIMED){ |  |  |  |  |
| media | Number | M | 0 | Index of the media in the MPEG media array used to retrieve the media playback timeline |
| timeLowerLimit | Number | M | 0 | Indicates the start time offset into the media playback timeline at which the trigger is activated, in second. The default value of 0 means the activation of the trigger at the start of the media playback. |
| timeUpperLimit | Number | O |  | Indicates the end time offset into the media playback timeline at which the trigger is deactivated, in second. If not present, the trigger is active until the end of the media timeline. |
| } |  |  |  |  |
| If(type == COLLIDER){ |  |  |  |  |
| nodes | Array | M |  | Indices of the nodes in the nodes array to be considered for collision determination. Any detection of collision shall activate the trigger |
| } |  |  |  |  |

Table 4 : Semantic of a trigger

* + 1. **Action semantic**

The semantic of an action is provided in Table 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M | ACTIVATE | Defines the type of the action by taking one of the following values:  ACTIVATE = 0,  TRANSFORM = 1,  ANIMATE = 2,  CONTROL\_MEDIA = 3,  PLACE\_AT = 4,  MANIPULATE = 5,  SET\_MATERIAL = 6 |
| delay | number | O |  | Duration of delay in second before executing the action |
| If(type == ACTIVATE){ |  |  |  |  |
| activationStatus | enum | M | ENABLED | ENABLED=0: the node shall be considered by the application  DISABLED =1: the node shall not be considered by the application |
| nodes | array | M | [] | Indices of the nodes in the nodes array to set the activation status |
| } |  |  |  |  |
| If(type== TRANSFORM){ |  |  |  |  |
| transform |  | M |  | 4x4 transformation matrix to apply to the nodes |
| nodes | array | M |  | Indices of the nodes in the nodes array to be transformed |
| } |  |  |  |  |
| If(type == ANIMATE){ |  |  |  |  |
| animation | number | M |  | index of the animation in the animations array to be considered |
| animationControl | enum | M | PLAY | PLAY = 0,  PAUSE = 1,  RESUME = 2,  STOP = 3 |
| } |  |  |  |  |
| If(type == CONTROL\_MEDIA){ |  |  |  |  |
| media | number | M |  | Index of the media in the MPEG media array to be considered |
| mediaControl | enum | M | PLAY | PLAY = 0,  PAUSE = 1,  RESUME = 2,  STOP = 3 |
| } |  |  |  |  |
| If(type == PLACE\_AT){ |  |  |  |  |
| placeDescription | string | M |  | Describe the place position. E.g. “/user/hand/left/pose” |
| nodes | array | M |  | Indices of the nodes in the nodes array to be placed. |
| } |  |  |  |  |
| If(type== MANIPULATE){ |  |  |  |  |
| action | enum | M | FREE | FREE= 0: the nodes follow the user pointing device and its rotation,  FREE\_FIXED\_ROTATION=1: the nodes follow the user pointing device but without rotation,  SLIDE=2: the nodes move linearly along the provided axis by following the user pointing device  TRANSLATE=3: the nodes translate by following the user pointing device,  ROTATE=4: the nodes rotate around the provided axis by following the user pointing device,  SCALE=5: performs a central scaling of the nodes by following the user pointing device |
| axis | array | O |  | (x,y,z) coordinates of the axis used for rotation and sliding. These coordinates are relative to the local space created by the USER\_INPUT trigger activation. E.g. a “/user/hand/left/pose” user input trigger creates a local space attached to the user left hand |
| nodes | array | M |  | Indices of the nodes in the nodes array to be manipulated |
| } |  |  |  |  |
| If(type == SET\_MATERIAL){ |  |  |  |  |
| material | number | M |  | Index of the material in the materials array to apply to the nodes |
| nodes | array | M |  | Indices of the nodes in the nodes array to set their material |
| } |  |  |  |  |

Table 5 : Semantic of an action

* + 1. **Behavior semantic**

The semantic of a behavior is provided in Table 6.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| triggers | array | M |  | Indices of the triggers in the triggers array considered for this behavior |
| actions | array | M |  | Indices of the actions in the actions array considered for this behavior |
| triggersControl | enum | M | LOGICAL\_OR | LOGICAL\_OR = 0: an activation of any of the defined triggers shall execute the defined actions,  LOGICAL\_AND=1: all the defined triggers shall be activated to execute the defined actions |
| actionsControl | enum | M | SEQUENTIAL | Defines the way to execute the defined actions.  SEQUENTIAL=0: each defined action is executed sequentially in the order of the actions array,  PARALLEL=1: the defined actions are executed concurrently |
| interruptAction | number | O |  | Index of the action in the actions array to be executed if the behavior is still on-going and is no more defined in a newly received scene update |
| priority | number | M |  | Weight associated to the behavior. Used to select a behavior when several behaviors are active at same time for one node |

Table 6 : Semantic of a behavior

* + 1. **Processing model**

During runtime, the application iterates on each defined behavior and checks the realization of the related triggers following the procedure detailed in Figure 6.



Figure 6 : Processing model to activate a trigger

When the defined triggers of a behavior are activated, then the corresponding actions are launched.

A behavior has an “*on-going*” status between the launch and the completion of its defined actions.

When several behaviors are in concurrence to affect the same node at the same time, the behavior having the highest priority is processed for this affected node. The other concurrent behaviors are then not processed.

Once achieved, the application iterates on each behavior as defined in Figure 6.

If a node is affected by concurrent behaviors with a same priority value, then the application shall manage the potential conflict.

When a new scene description update is received, the application shall follow the procedure detailed in Figure 7.

A behavior is considered on-going when the related action(s) is(are) currently being executed when the scene update is processed.

A behavior is considered “*still defined*” if its unique association of (triggers, actions) is still described in the scene update.

If a behavior is no more “*still defined*”, its interrupt action is executed.

When all the interrupt actions (if any) are achieved, then the application removes any obsolete scene data and considers any new data to match the updated scene description.



Figure 7 : Processing model when a new scene description is received

* 1. **Test cases**

The EE may define test cases for which the evaluation criteria will be analyzed. For instance, a first test case can be with live content while another in the on-demand content.

* 1. **Evaluation criteria**

List of criteria that will allow to compare the different technical solutions and converge to a unique solutions. Criteria can be objective like memory efficiency, bitrate or subjective flexibility, compatibility with legacy solution, etc..

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Description** | **Evaluation** |
| Criteria #1 | Description | The technical solution should minimize/optimise … |

* 1. **Timeline**
* 2022-01-17: MPEG #138: Refine architecture and define basic triggers and actions
  1. **References**

[1] “Technologies under Consideration on Scene Description for MPEG Media”, N00367, MPEG2021, Online, October 2021

1. **EE6: User Representation and Avatars (new)**
   1. **Introduction**

The MPEG Scene Description group relies on the glTF2.0 technology to enable the support of 3D scene in MPEG media. Using the extension mechanism, the solution allows synchronization between traditional MPEG media within 3D content.

As defined in the requirements, one goal is to permit a user to navigate the content and interact with the surrounding objects and characters [1].

* 1. **Problem statement**

In order to interact within the 3D scene, the user must be represented in the scene. This representation is called an avatar and reinforces the user’s feeling of presence in the virtual world. An avatar is not mandatory if the user is simply walking through and watching some content, but as soon as there is interactivity and collision, the user must be able to visualize or detect the boundaries of the avatar. As of today, the representation of the user within the scene is not formally defined. Requirements for MPEG-I Phase 2 only mention the user as “the listener whose position and orientation are used for rendering” [2].

This EE will focus on developing the necessary glTF extensions to support user representations in scene description.

* 1. **Use cases relevant for the EE**

Basically, all use cases listed by the haptic group require an avatar so the user can touch virtual objects [3]. The audio use cases do not explicitly mention the user appearance although the objects in the scene impact the sound rendering (#4) [5]. Besides the social VR scenario implicitly means that users can see each other's.

* 1. **Related (WG2) and Extracted (new) Requirements**

The following requirements are thus relevant and addressed by this EE:

* Support of user interactivity with objects within a virtual environment (requirement #90)
* Support of interaction between multiple users within an immersive environment (requirement #95)
* The specification shall support interactivity models related to avatar position and orientation (requirement #129)
* The specification shall support coding and presentation of interactivity models related to avatar-scene or avatar-avatar interactions (requirement #130)
* The specification shall support different media types and various haptic feedback paradigms (requirement #131)
  1. **Relation to other activities (EE, requirements, etc…)**

A relationship to the Haptics phase 2 activity has been identified. The user could touch virtual objects, hence a visual representation is mandatory.

This EE relates with the MPEG Systems EE on Interactivity [4] and Haptics [6].

* 1. **Mandates**

The mandates for this EE are as follows:

* Define the term avatar in the MPEG-I Phase 2 requirements
* Identify the existing glTF-based solutions to describe avatars
* Define the scope of the glTF extension for avatars within the scene description architecture
* Define test scenarios and collect test assets
* Evaluate proposed solutions
* Develop the reference software integration and validate against the test scenarios
  1. **Participants**

|  |  |  |  |
| --- | --- | --- | --- |
| Participant | Contact | Email | Type |
|  |  |  |  |
| InterDigital | Fabien Danieau | [fabien.danieau@interdigital.com](mailto:fabien.danieau@interdigital.com) | L |
| Qualcomm | Imed Bouazizi | bouazizi@qti.qualcomm.com | P |
| Immersion | Yeshwant Muthusamy | ymuthusamy@immersion.com | P |

(P = proponent, L = leader)

* 1. **Information about proposed technologies**

List of already submitted contributions on this topic.

* + 1. **m56337 [SD] Interactivity in Scene Description**

This contribution introduced the user as the camera controller which is his/her only representation in the scene.

* + 1. **m58104 [SD] On scene interactivity**

This contribution presented the camera as the user avatar. It also presents a need for a collider so the user cannot walk beyond the limited space of the experience.

* + 1. **m58146 [SD] Describing camera paths for interactivity**

This contribution also considers the user as the camera and limits his/her movement to a camera path.

* + 1. **m58487 [SD] MPEG-I SD Revised Haptic Schema and Processing Model**

This contribution discusses the problem of the user representation and mentions a potential solution VRM to be evaluated. it is also indicated that the haptic needs are more focused on the collision (i.e. bounding box) than the visual appearance.

* 1. **Test cases**

The EE may define test cases for which the evaluation criteria will be analyzed. For instance, a first test case can be with live content while another in the on-demand content.

* + 1. **Camera representation**

This test case corresponds to the use case proposed by m56337 and m58104. It has no visual appearance but only controls the camera viewport. A collider is associated to this “body representation” so the user cannot go beyond a limited area. This collider will also be used to detect was body part collides with an object and thus trigger a haptic effect.

* + 1. **Limited representation**

Same as above, with the display of the user’s head and hands (most current VR avatars). Tracking is performed by a VR headset and associated controllers. User can interact with objects in the scene.

* + 1. **Full body avatar**

Same as above, with the display of the full body avatar. Full body tracking is required or simulated. Multiple users can see each other.

* 1. **Evaluation criteria**

List of criteria that will allow to compare the different technical solutions and converge to a unique solutions. Criteria can be objective like memory efficiency, bitrate or subjective flexibility, compatibility with legacy solution, etc..

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Description** | **Evaluation** |
| #1 mesh | A mesh represents the avatar | The mesh is compliant with glTF format |
| #2 collider | A collider surrounds the avatar | The collider allows to detect precise collision on the user’s body |
| #3 bones | The mesh is rigged | The set of bones is formally defined and allows motion tracking |

* 1. **Timeline**
* 2022-01-17: MPEG #138
  + Define the term avatar
  + Define the scope of the avatar extension
  + Analyze existing solutions
  + Propose test cases and criteria
* 2022-07-18: MPEG #139
  + completion of EE
  1. **References**

1. “Requirements Coverage of MPEG-I Scene Description”, N00369, MPEG2021, Online, October 2021
2. “Requirements for MPEG-I Phase 2 WG 2, MPEG Technical requirements”, m57684, MPEG2021, Online, July 2021
3. “[Haptics] Updated MPEG-I Phase 2 Haptics Use Cases”, m57952, MPEG2021, Online, October 2021
4. “[SD] Description of EE on Generic Interactivity Framework”, m59166, MPEG2021, Online, January 2022
5. “Thoughts on MPEG-I Audio requirements”, m46062, MPEG2019,Marrakesh, MA, January 2019

“[SD] Description of EE on Haptics Support in SD”, m59210, MPEG2021, Online, January 202

1. **EE7: AR Anchoring (new)**
   1. **Introduction**

AR is a technology that offers immersive experiences that cover a wide range of applications such as gaming and conferencing. The user fuses virtual reality with their own reality to build the immersive experience. Scene description is a tool that can be used to describe the virtual 3D scene that will be used to augment the user’s reality.

This EE will evaluate different solutions to anchor a scene into the user’s environment to produce the augmented reality experience.

* 1. **Problem statement**

The following aspects of AR anchoring will be studied:

* means to spatially anchor a scene to a user’s environment
* adjustment of the scene scale to match the user’s environment
* identification of anchor points and planes for the scene
  1. **Use cases relevant for the EE**

In this simple use case, the user sits in her living room and uses their tablet to watch a football game. The user clicks on a button to activate the AR mode on the tablet or transfer the game to their AR glasses, so that they can now watch the game on top of their living room table. They can move around the table to see the game from different angles as shown in the following picture.

A person playing a game

Description automatically generated with medium confidence

Different variants of this use case are possible.

* 1. **Related (WG2) and Extracted (new) Requirements**

The following requirements are relevant for this EE:

* The spatial description shall contain XR features for indoor and/or outdoor localization/tracking purpose (requirement #139)
* The spatial description should have a data structure to represent XR space with appropriate pixel-to-meter scaling factor to the physical world (requirement #140)
* The spatial description shall allow the definition of the XR space from small scale space like indoor environment to large scale outdoor space (requirement #141)
* The spatial description shall support visual and/or non-visual features (including audible and mechanical features) of the XR space for real-time localization and tracking (requirement #142)
* The spatial description shall allow navigation of the XR client from one space to another when multiple indoor XR spaces are used (requirement #143)
* The XR spatial description shall include XR anchor objects with presentation properties as the placement of XR media object (requirement #144)
* The XR spatial description should support stationary and moving (mobile) XR anchors (requirement #145)
* The spatial description should keep a mapping between virtual coordinate system to one or multiple physical coordinate systems for different space (requirement #146)
* The position for media object in the XR space shall be defined by an XR anchor (requirement #147)
* There shall be an association defined logic links between XR anchors and live streaming media object (requirement #148)
* There shall be an association defined for logic links between XR anchors and on demand media object (requirement #149)
  1. **Relation to other activities (EE, requirements, etc…)**

This EE is related to the EE on the generic interactivity framework.

* 1. **Mandates**

The mandates for this EE are as follows:

* evaluate the solutions for scene anchoring and scene scaling
* document the relevant use cases and specify the validation scenarios
* develop the reference software integration for the selected scene anchoring solution
* validate the solution against the identified validation scenarios
* study the problem of defining and signalling anchor points/planes and determine what needs to be signalled as part of the scene description
  1. **Participants**

|  |  |  |  |
| --- | --- | --- | --- |
| Participant | Contact | Email | Type |
|  |  |  |  |
| Qualcomm | Imed Bouazizi | bouazizi@qti.qualcomm.com | L |
| Interdigital | Fabien Danieau | fabien.danieau@interdigital.com | P |
| Interdigital | Patrice Hirtzlin | patrice.hirtzlin@interdigital.com | P |
| Interdigital | Gurdeep Bhullar | gurdeep.bhullar@interdigital.com | P |
| Xiaomi | Emmanuel Thomas | thomase@xiaomi.com | P |

(P = proponent, L = leader)

* 1. **Information about proposed technologies**

The following contributions on AR anchoring have been identified:

### **MPEG Meeting #134**

[m56781](https://dms.mpeg.expert/doc_end_user/documents/134_OnLine/wg11/m56781-v1-m56781.zip) [SD] Support for AR in Scene Description

### **MPEG meeting #135**

[m57408](https://dms.mpeg.expert/doc_end_user/documents/135_OnLine/wg11/m57408-v2-m57408.zip) [SD] AR support in scene description

### **MPEG meeting #136**

[m58122](https://dms.mpeg.expert/doc_end_user/documents/136_OnLine/wg11/m58122-v2-m58122_AR_anchors_REVISED_with_tracked_changes.zip) [SD] AR anchors in MPEG-I Scene Description

[m58489](https://dms.mpeg.expert/doc_end_user/documents/137_OnLine/wg11/m58489-v1-m58489.zip) [SD] OpenXR backend in SD reference software

### **MPEG meeting #137**

[m58505](https://dms.mpeg.expert/doc_end_user/documents/137_OnLine/wg11/m58505-v1-m58505_SD_XR_Anchors.zip) [SD] On alternative anchoring scheme

[m58869](https://dms.mpeg.expert/doc_end_user/documents/137_OnLine/wg11/m58869-v1-m58869.zip) [SD] On anchors integration in SD

[m58922](https://dms.mpeg.expert/doc_end_user/documents/137_OnLine/wg11/m58922-v1-m58922.zip) [SD] Updates on AR scene anchoring

* 1. **Test cases**

To be defined

* 1. **Evaluation criteria**

The evaluation criteria are to be defined.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Description** | **Evaluation** |
| Criteria #1 | Description | The technical solution should minimize/optimise … |

* 1. **Timeline**
* 2022-01-17: MPEG #138: Refine architecture and define basic triggers and actions
  1. **References**

[1] “Technologies under Consideration on Scene Description for MPEG Media”, N00367, MPEG2021, Online, October 2021

1. [MPEG/Systems/SceneDescription/MPEG-Contributions#222](file:///\\code1\software\MPEG\Systems\SceneDescription\MPEG-Contributions\-\issues\222) [↑](#footnote-ref-1)