**Text

Description automatically generatedISO/IEC JTC 1/SC 29/WG 03 N01734**

**ISO/IEC JTC 1/SC 29/WG 03  
MPEG Systems   
Convenorship: KATS (Korea, Republic of)**

**Document type:** Output document

**Title:** Draft text of ISO/IEC 23090-14 2nd edition FDAM 1: Support of MPEG-I immersive audio, scene understanding and other extensions

**Date of document:** 2026-02-05

**Source:** WG03

**No. of pages:** 29 (with cover page)

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

ISO/IEC 23090-14:XXXX/DAM 1:2025(en)

ISO/IEC JTC1/SC 29

Secretariat: JISC

Date: 2025-05-16

Information technology — Coded representation of immersive media — Part 14: Scene description — Amendment 1: Support of MPEG-I audio, scene understanding and other extensions

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

CP 401 • Ch. de Blandonnet 8

CH-1214 Vernier, Geneva

Phone: +41 22 749 01 11

Fax: +41 22 749 09 47

Email: copyright@iso.org

Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](https://www.iso.org/directives-and-policies.html) or [www.iec.ch/members\_experts/refdocs](https://www.iec.ch/members_experts/refdocs)).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at [www.iso.org/patents](http://www.iso.org/patents) and [https://patents.iec.ch](https://patents.iec.ch/iec/pa.nsf/pa_h.xsp?v=0). ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see [www.iso.org/iso/foreword.html](https://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](https://www.iec.ch/understanding-standards).

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

A list of all parts in the ISO/IEC 23090 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](https://www.iso.org/members.html) and [www.iec.ch/national-committees](https://www.iec.ch/national-committees).

**Information technology — Coded representation of immersive media — Part 14: Scene description — Amendment 1: Support of MPEG-I audio, scene understanding and other extensions**

*8.1.1*

Add the following paragraph, after the first paragraph:

Once anchored, the elements of the scene may interact with the physical environment. The anchoring extension also specifies how the representation of this environment may be provided to manage these interactions and to reach expected AR experience.

*8.1.2, Table 28*

Add the following row at the bottom of Table 28:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| recommendedSpatialComputingConfig | object | O | N/A | Provides a set of recommended parameters specifying the needed spatial description. The semantics of this object are given in Table 59. |

Add the following table after Table 28:

Table  59 — RecommendedSpatialComputingConfig Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| scanOptions | array | O | N/A | Array of options (enumeration) for the scan computation: possible values are given in  Table 8.1- 2. |
| scanDetails | object | O | N/A | Specifies the required level of detail for the representation. The semantics are presented in Table 8.1- 3. |
| scanUpdate | object | O | N/A | Specifies the frequency at which the spatial description must be updated (Table 8.1- 4). |
| scanVolumes | array | O | N/A | Array of bounding volumes that determine the spaces where scanned objects must be provided. Real scan objects that intersect one or more of the bounding volumes should be provided, and all other objects ignored. The semantics for these volumes are presented in Table 8.1- 5. |
| realSemantic | array | O | N/A | Semantic descriptions of nodes that are needed (“table”, “room”, “chair”, “wall”, “light”, “freespace” …) |
| lightOptions | array | O | N/A | Array of option (enumeration) for the light extraction: possible values are given in Table 8.1- 6. |
| lightUpdate | object | O | N/A | Specifies the frequency at which the extraction of real light must be updated (Table 8.1- 4) |

Table 60 — Possible values for a scanOptions item

|  |  |
| --- | --- |
| **Enumeration value** | **Description** |
| PLANE = 0 | Request plane data for scanned objects |
| PLANAR\_MESH | Request planar meshes for scanned objects |
| VISUAL\_MESH | Request 3D visualization meshes for scanned objects |
| COLLIDER\_MESH | Request 3D collider meshes for scanned objects |
| FREE\_VOLUME | Request to get the available space around a trackable |
| BOUNDING\_BOX | Request a simplified collider mesh |
| TEXTURED\_MESH | Request mesh with a texture |

Table 61 — Semantics for the scanDetail object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| primitivesNumber | number | O | N/A | Gives the quantity of geometric primitives per cubic meter (m3) |
| textureOption | enumeration | O | 0 | Gives the resolution of the texture of textured meshes. Possible values are R512X512 (0), R1024X1024 (1) or R2048X2048 (2). |

Table 62 — Semantics of a *scanUpdate* or *lightUpdate* object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| occurrence | enumeration | O | ONCE | An occurrence value: |
|  |  |  |  | — ONCE = 0: the update is performed only once, for instance in case of a static real scene,  — N\_FRAME: the update is performed periodically, every N rendering frames, N depending on the dynamism of the real scene. The N value is provided in the *frameNumber* parameter.  — AUTO: the update frequency is managed by the module that computes the representation. This module may perform an analysis to detect significant changes in the real world and start an update. This analysis may be performed from raw images data, like RGB images from a camera or depth images from a depth sensor. |
| numberOfFrames | number | O | N/A | Indicates the periodicity, in number of frames, of the update, when the *occurrences* value is N\_FRAME. |

Table 63 — semantics for a scanVolumes item

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M |  | SPHERE=0, BOX, FRUSTUM |
| If (type == SPHERE) |  |  |  |  |
| center | array | M |  | 3D coordinates of the center of the sphere |
| radius | number | M |  | Radius of the sphere in meters. |
| If (type == BOX) |  |  |  |  |
| pose | matrix | M |  | 4\*4 matrix representing the center position and orientation of the box |
| extents | array | O |  | Edge-to-edge length of the box along each dimension. |
| If (type == FRUSTUM) |  |  |  |  |
| pose | matrix | M |  | 4\*4 matrix representing the position and orientation of the tip of the frustum |
| fov | vec4 | M |  | Angles of the four sides of the frustum |
| far | number | M |  | Positive distance of the far plane of the frustum |
| near | number | M |  | Positive distance of the near plane of the frustum |
| 3D coordinates are expressed in the XR space related to trackable associated to the anchor. | | | | |

Table 64 — Possible values for a lightOptions item

|  |  |
| --- | --- |
| **Enumeration value** | **Description** |
| DirectionalLight = 0 | Request the extraction of directional lights |
| EnvLight | Request the extraction of environment lights |
| PointLight | Request the extraction of point lights |
| SpotLight | Request the extraction of spot lights |
| AreaLight | Request the extraction of area lights |

*8.1.3*

Add the following paragraphs at the end of the subclause:

When processing the MPEG\_anchor extension, if a ***recommendedSpatialComputingConfig*** objectis present, the Presentation Engine checks if it can retrieve the recommended spatial description, specified in the parameters such as scanOptions, scanDetails, scanUpdate, scanVolumes, scanSemantic, lightUpdate and lightOptions. If all the recommended parameters are not satisfied, the Presentation Engine may continue the rendering of the scene with the available spatial description, but possibly with a degraded XR experience.

At runtime, the Presentation Engine may then request elements of the spatial description according to the ***recommendedSpatialComputingConfig*** parameters***.***

*8.2.1*

Add the following paragraph at the end of the subclause:

When specified, at node and/or scene level, a physics object specifies a set of parameters for the physic engine, used by the application.

*8.2.2.1, Table 32*

Replace Table 32 with the following table:

Table 32 — Semantic of the MPEG\_scene\_interactivity extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| physics | object | O | N/A | Provides a set of parameters at scene level to be used for the physics simulation. The semantics of this object are given in table 8.2-1. |
| 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 |

Add the following text and new table after Table 32:

The semantics of a *physics* object at scene level, are defined in Table 65.

Table 65 — Semantics of a scene level physics object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| recommendedPhysicsHighPrecision | boolean | O | false | Determines whether the application should enable a more deterministic and precise physic simulation |
| gravity | number | O | -9.81 | Determine the gravity for the whole scene. In meters per secondsquare (m.s-2), as defined in the international unit system. |
| recommendedPhysicsFrameRate | number | O | 50 | Provides the recommended frame rate at which the Physics Engine should operate. In frame per second, as defined in the international unit system. |
| bounceThreshold | number | O | 1 | A contact with a relative velocity below this threshold will not result in a bounce. In meter per second (m.s-1), as defined in the international unit system. |

*8.2.2.1*

Add the following text before Table 33:

The annex I gives extra semantics for user input triggers involving multiple users.

*8.2.2.2, Table 52*

Replace Table 52 with the following table:

Table 52 — Semantic of the MPEG\_node\_interactivity.trigger extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M |  | One element of Table 34 that defines the type of the trigger. |
| if (type == TRIGGER\_COLLISION){ |  |  |  |  |
| collider | integer | M |  | the index of the mesh element that provides the collider geometry for the current node.  The collider mesh may reference a material. |
| isStatic | boolean | M |  | If True, the collider is defined as a static collider. |
| physics | object | O | N/A | Provides a set of parameters at node level to be used for physics simulation. The semantics of this object are given in table 66. |
| primitives | array(Primitive) | O | N/A | List of primitives used to activate the proximity or collision trigger.  Semantics are presented in Table 35. |
| } |  |  |  |  |
| … |  |  |  |  |

Add the following text and new table after Table 52:The semantics of a *physics* object at node level, are defined in Table 66.

Table 66 — Semantics of a node level physics object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| needPreciseCollisionDetection | boolean | O | false | If true, the physics engine should handle the collision detection more accurately by increasing the detection rate for this node. |
| linearDamping | number | O | 0 | A non-negative value, in second-1 (s-1), as defined in the international unit system. It defines the linear drag coefficient which corresponds to the rate of decrease of the linear velocity over time.    It is used to compute a new velocity value V(t) at each simulation step (dt):    V(t+dt) = V(t)\*(1-linearDamping\*dt), the velocity being clamped to 0. |
| angularDamping | number | O | 0 | A non-negative value, in second-1 (s-1), as defined in the international unit system. It defines the angular drag coefficient which corresponds to the rate of decrease of the angular velocity over time.    It is used to compute a new velocity value V(t) at each simulation step (dt):    V(t+dt) = V(t)\*(1-angularDamping\*dt), the velocity being clamped to 0. |
| useGravity | boolean | M |  | Indicates if gravity affects the object |
| mass | number | M |  | Mass of the object in kilogram, as defined in the international unit system. |
| restitution | number | M |  | Provides the ratio of the final to initial relative velocity between two objects after they collide |
| staticFriction | number | M |  | Unitless static friction coefficient as defined in the Coulomb friction law [10]. Friction is the quantity which prevents surfaces from sliding off each other. Static friction is used when the object is lying still. It will prevent the object from starting to move. |
| dynamicFriction | number | M |  | Unitless static friction coefficient as defined in the Coulomb friction law. When a large enough force is applied to the object, a dynamic friction is used and will attempt to slow down the object while in contact with another. |
| } |  |  |  |  |

*8.2.3, last two paragraphs*

Replace the last two paragraphs with the following:

The application shall handle a physics simulation if a *physics* object is defined at a scene level or/and at any of the nodes. When a collision occurs between two nodes, the application should calculate the combination of the restitution, static friction and dynamic friction values based on the values provided by the physics objects of the two nodes.

*8.4.2.2*

Replace the second paragraph and table title with the following:

The definition of all objects within the MPEG\_lights\_punctual extension is provided in Table 58.

Table 58 —Definition of glTF file objects MPEG\_lights\_punctual extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| lights | array | M |  | An array of items that describe the punctual light sources, referenced in this scene description document. The semantics of each item is specified in Table 67 |

Table 67 —Definition of item in the lights array of the MPEG\_lights\_punctual extension

*A.11*

Replace:

MPEG\_anchor.anchor schema is downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/MPEG_anchor/schema/MPEG_anchor.anchor.schema.json>.

With:

MPEG\_anchor.anchor schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_anchor.anchor.schema.json>.

MPEG\_anchor.anchor.recommendedspatialcomputingconfig schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_recommendedspatialcomputingconfig.schema.json>.

*A.13*

Replace:

MPEG\_scene\_interactivity schema is downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/MPEG_interactivity/schema/MPEG_interactivity.schema.json>.

With

MPEG\_scene\_interactivity schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_interactivity.schema.json>.

MPEG\_scene\_interactivity.physics schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_interactivity.physicsobject.schema.json>.

MPEG\_scene\_interactivity.trigger.multiuser schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_interactivity.trigger.multiuserobject.schema.json>.

Replace:

MPEG\_node\_interactivity schema is downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/MPEG_interactivity/schema/MPEG_node_interactivity.schema.json>.

With:

MPEG\_node\_interactivity schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_node_interactivity.schema.json>.

MPEG\_node\_interactivity.physics schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_node_interactivity.physicsobject.schema.json>.

*A.15*

Replace:

MPEG\_lights\_texture\_based schema is downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/MPEG_lighting/schema/MPEG_lights_texture_based.schema.json>.

With:

MPEG\_lights\_texture\_based schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_lights_texture_based.schema.json>.

*A.16*

Replace:

MPEG\_light\_punctual schema is downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/MPEG_lighting/schema/MPEG_lights_punctual.schema.json>.

With:

MPEG\_lights\_punctual schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_lights_punctual.schema.json>.

MPEG\_lights\_punctual.light schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_lights_punctual.light.schema.json>.

*Annex A,*

Add the following new clause as follows:

**A.19   JSON schema for MPEG\_node\_mapping**

MPEG\_node\_mapping schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_node_mapping.schema.json>.

*Annex A,*

Add the following new clause as follows:

**A.20   JSON schema for MPEG\_gaussian\_splatting\_transport**

MPEG\_gaussian\_splatting\_transport schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_gaussian_splatting_transport.schema.json>.

*Annex A,*

Add the following new clause as follows:

**A.21   JSON schema for MPEG\_material\_stereo**

MPEG\_material\_stereo schema is available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/schemas/MPEG_material_stereo.schema.json>.

*Annex F,*

*F.1*

Replace:

In the example downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/example_MPEG_media.json>, two media items are listed by MPEG\_media extension.

With:

In the example available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/examples/example_MPEG_media.json>, two media items are listed by MPEG\_media extension.

*F.4*

Replace:

In the example downloadable from <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/example_MPEG_scene_dynamic.json>, the media object includes the patch document format file name and its track index.

With:

In the example available at: <https://standards.iso.org/iso-iec/23090/-14/ed-2/en/amd/1/examples/example_MPEG_scene_dynamic.json>, the media object includes the patch document format file name and its track index.

*Annex G*

Add the following new Clause as follows:

**G.4   Support for MPEG-I immersive audio**

**G.4.1   General**

MPEG-I Immersive Audio has been specified in ISO/IEC 23090-4. The specification assumes the presence of an MPEG-I immersive audio renderer that will receive the MPEG-I immersive audio bitstream, a set of MPEG-H audio streams, as well as information about some scene metadata, such as listener’s pose. It will then use the audio scene metadata in the MPEG-I immersive audio bitstream, the decoded audio data, for example, from MPEG-H 3D audio (ISO/IEC 23003-3) streams, and the pose information to render the spatial audio. In case of coded audio data with any other codec, the respective decoder is required.

The support of MPEG-I immersive audio is achieved by referencing an MPEG-I immersive audio stream in a MPEG-I scene description document.

The MPEG-I immersive audio bitstream contains a description of the audio scene that is independent of the main scene description consumed by the Presentation Engine. An alignment between the Presentation Engine and the MPEG-I immersive audio Renderer is needed, that goes beyond the traditional time alignment but includes also spatial alignment. For that, a mapping need to be established between the node in a MPEG-I scene description document and the node of the audio scene.

The following figure depicts an example of a mapping between a node that contains a car and an external audio node in an MPEG-I immersive audio bitstream, with a simplified geometry of that car and the attached audio sources. This mapping is described in an MPEG node mapping extension.

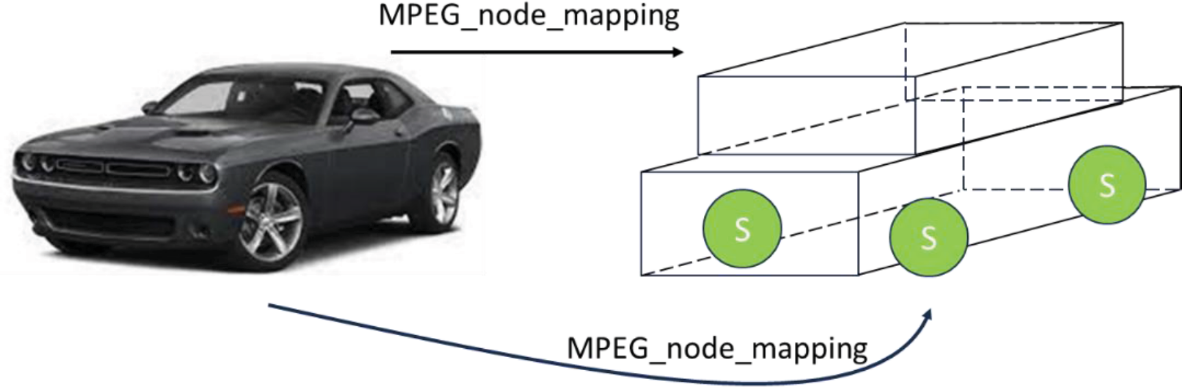


Figure G.4 — 1: MPEG-I audio mapping example

This MPEG node mapping extension, identified by MPEG\_node\_mapping, can be used in a broader scope: It establishes a mapping between the node in a MPEG-I scene description document and an external entity, e.g. an MPEG-I immersive audio renderer, that handles a dedicated scene graph, separate from the main scene description. When present, the MPEG\_node\_mapping extension shall be included in a node object.

The architecture for the support of such an external renderer is depicted in the following figure:

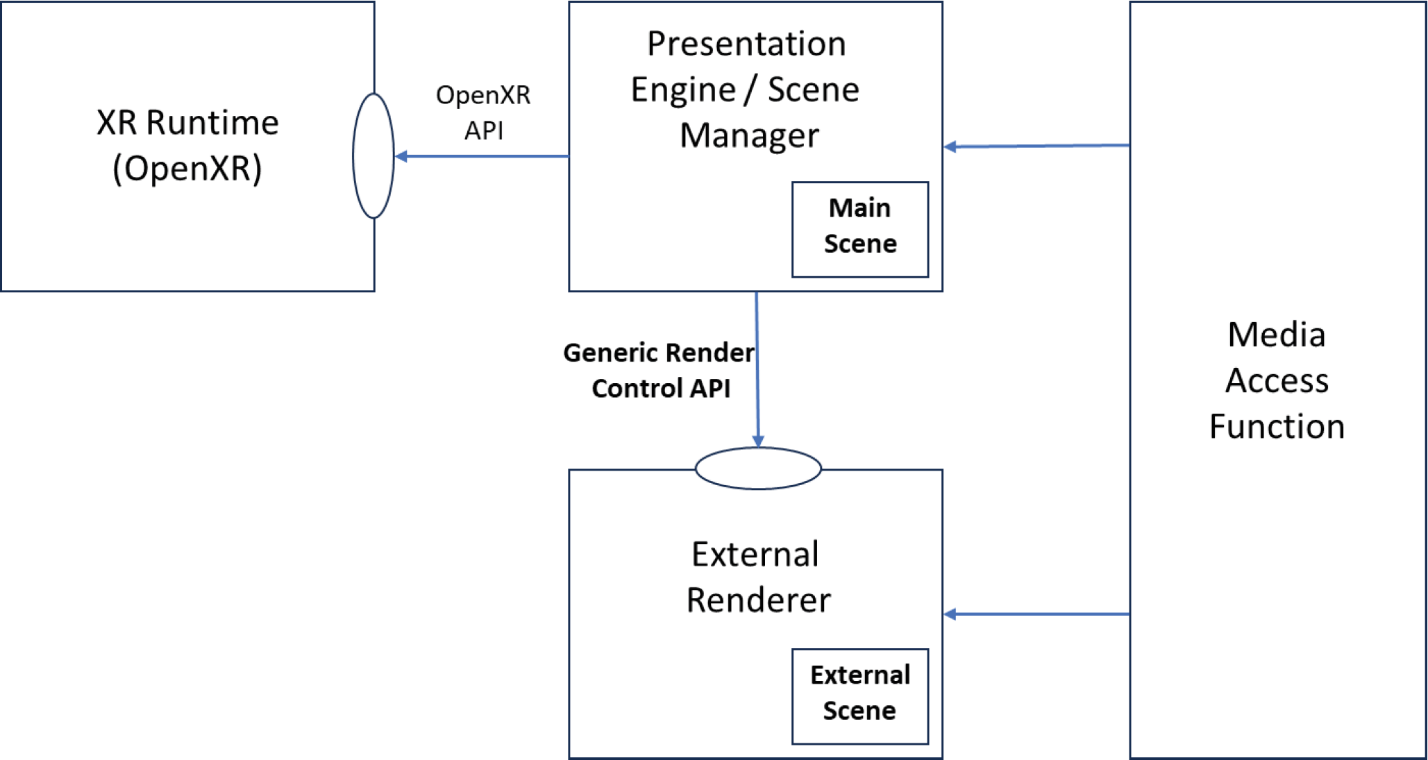


Figure G.4 — 2: Architecture for external render support

The Generic Render Control API is an abstract API that is offered by external renderers to enable applications, such as Presentation Engines, to control the rendering process by aligning and synchronizing their rendering state to that of the Presentation Engine. This API is used by the Presentation Engine to configure and update the status of the external renderer.

**G.4.2   Generic Render Control API**

Table G.21 describes the functionality provided by the generic render control API:

Table 21 — Generic Render Control API

| **Method** | **Description** | |
| --- | --- | --- |
| init() | Initializes the external renderer by providing the related media source information and their corresponding buffers. It also establishes a session between the Presentation Engine and the external renderer.    The inputs to this method call should be: | |
|  |  | — A media source object that contains a handler to the buffer(s), where the source media will be made available by the MAF. A description of the media source and the contents of each buffer shall also be provided. |
| configure() | Configures the external renderer to establish an initial alignment and synchronization between the Presentation Engine and the external renderer.  The parameters to this method may include: | |
|  |  | — A mapping between the initial timestamp of the common Presentation Engine timeline and that of the media associated with the external renderer. It also provides information about the clock rate of the Presentation Engine.  — A list of mapped nodes in the source media rendered by the external renderer. This list shall at least contain one object with a mapping to the main camera of the main scene description. For audio renderers, this may be the audio listener. The information is provided by the MPEG\_node\_mapping extension in the scene description document. It should also provide the initial pose of the mapped nodes after applying the 4x4 matrix of the *transform* parameter provided in the node mappings. |
|  | The external renderer may then subscribe for updates to specific aligned nodes, or it may specifically ask for current state for these nodes, using the referenceId. | |
| start()  pause()  resume()  stop() | Allows the Presentation Engine to control the playback of selected media sources associated with the external renderer for interactivity purposes. | |
| update() | Used by the Presentation Engine to update node positions and orientations for which there is a mapping with the external renderer. Updates may result from received scene updates, user interactions, animations, physics simulations, or any other events. The Presentation Engine uses this update() method when one or more mapped nodes need to be spatially synchronized, depending on their *synchronizationOccurrence* and s*ynchronizationOccurenceCombination* parameters provided in their MPEG\_node\_mapping extension.  The parameters passed to this method are an array of objects consisting of: | |
|  |  | — The referenceId of the node to which this update applies  — The current pose of the mapped node after applying the 4x4 matrix of the *transform* parameter provided in the corresponding MPEG\_node\_mapping |
| updateGraph() | The Presentation Engine uses the updateGraph function to add, update, or remove a set of nodes to the internal representation of the scene that is maintained by the external renderer. Only nodes that have a mapping with the external renderer can be passed through this method.  The parameters to this method are an array of objects that include: | |
|  |  | — The graph operation: ADD, REMOVE, UPDATE  — For ADD: the referenceId and the initialization information for the associated media data to the object that is to be added.  — For REMOVE: the referenceId of the object to be removed.  — For UPDATE: the referenceId of the object to be updated, as well as a dictionary of attributes and their update values. |
| registerCallback() | The Presentation Engine may provide a callback function to the external renderer to allow it to query the status of certain parameters at any time. This may for example include asking for the current user pose.    The Presentation Engine shall register a callback function whenever possible. | |

The following is a description for the API in Interface definition language (IDL, ISO/IEC 19516):

|  |
| --- |
| interface GenericRenderControl {     void init();    void configure();  void start();  void pause();  void resume();  void stop();  update();  void updateGraph();    void registerCallback(); }; |

**G.4.3   Semantics**

The semantics of the MPEG\_node\_mapping extension are provided in the following table. When present, the MPEG\_node\_mapping extension shall be included in a node object.

Table G.22 — Semantics of the MPEG\_node\_mapping extension

| **Name** | **Type** | **Default** | **Usage** | **Description** |
| --- | --- | --- | --- | --- |
| mappings | array(object) |  | M | An array of mappings associated with the containing node. |
| component | string | “urn:mpeg:sd:component:default” | O | An identifier of the component associated with this mapping. The component may for instance be  “urn:mpeg:sd: component:audio-renderer” to indicate that the component is an audio renderer. |
| source | number | N/A | M | The index in the MPEG\_media that provides the media resource that contains the mapped element. |
| referenceId | number | N/A | M | An identifier of the element in the referenced resource. |
| transform | array(number) | Identity | O | A 4x4 TRS matrix which transforms the 3D coordinates of the node having this glTF extension expressed in the glTF2.0 scene coordinate system to the 3D coordinates of the node of the external renderer graph referenced by the referenceId identifier expressed in the external renderer scene coordinate system.  If the mapped node is a child of an AR Anchor/Trackable, the 4x4 TRS matrix transforms the 3D coordinates of the node having this glTF extension expressed in the AR Anchor/Trackable coordinate system to the 3D coordinates of the node of the external renderer graph referenced by the referenceId identifier expressed in the AR Anchor/Trackable coordinate system. |
| updateRecommendation | object | N/A | O | Indicate update recommendations for the node. The semantics is given in table x2. |
| supportsInteractivity | boolean | false | O | Indicates if interactivity actions applied to the node should be exposed if an API is made available to the Presentation Engine by the renderer of the resource. |

Table G.23— Semantics of an updateRecommendation object

| **Name** | **Type** | **Default** | **Usage** | **Description** |
| --- | --- | --- | --- | --- |
| synchronizationOccurrences | array(enumeration) | [EVENT] | O | An array of synchronization occurrences. Each element of this array is an Enumerator with the following possible values: |
|  |  |  |  | — ONCE: the synchronization is done once at the configuration step,  — EVENT: the synchronization is done based on the activation of one or more triggers (e.g., visibility, proximity, collision, user input) as those defined in the MPEG\_interactivity extension. The indices of the triggers, from the triggers array of the MPEG\_scene\_interactivity extension, are given in the *events* parameter,  — N\_FRAME: the synchronization is periodic every N (1, 2, …) rendering frames. The N value is provided in the *frameNumber* parameter. |
| events | array(number) | N/A | 0 | Array of indices of triggers from the triggers array of the MPEG\_scene\_interactivity extension. Required when EVENT is mentioned in the *synchronizationOccurrences* array. |
| frameNumber | number | N/A | 0 | Indicate the periodicity, in number of frames, of the synchronisation, when N\_FRAME is mentioned in the *synchronizationOccurrences* array. |
| synchronizationOccurrenceCombination | string | “|” | O | A set of logical operations to apply to the synchronization occurrences. A ‘#’ indicates the occurrence index, ‘&’ indicates a logical AND operation, ‘|’ a logical OR operation and ‘~’ a NOT operation. Parenthesis are used to group some operations. Such a syntax may give the following string: “#1&~#2|(#3&#4)”. |

**G.4.4   Processing model**

When processing the MPEG\_node\_mapping extension, the Presentation Engine shall identify nodes in the scene description that have a node mapping. The Presentation Engine shall determine if the component identified by the indicated component parameter supports the Generic Render Control API. If it does, the Presentation Engine shall pass the mapping information to the identified component.

The Presentation Engine shall then use the API to align the rendering with the component as configured over the API.

*Annex G*

Add the following new Clause as follows:

## G.5 Support for Gaussian Splats

### G.5.1 Overview

Gaussian splats are represented in MPEG-I Scene Description using the KHR\_gaussian\_splatting extension as the base layer, and an MPEG extension that extends KHR\_gaussian\_splatting for transport and streaming features.

The MPEG extension name shall be MPEG\_gaussian\_splatting\_transport.

The following signaling rules for glTF assets that carry Gaussian splats shall apply:

* The glTF asset shall list KHR\_gaussian\_splatting in extensionsUsed.
* The glTF asset shall list MPEG\_gaussian\_splatting\_transport in extensionsUsed when any MPEG-specific progressive or timed transport feature is used.
* The mesh primitive mode shall be POINTS.
* The MPEG\_gaussian\_splatting\_transport object shall be carried inside the nested extensions object of KHR\_gaussian\_splatting.

### G.5.2 Semantics

Table G.24 summarizes the primitive attribute semantics used for Gaussian splats (glTF + KHR\_gaussian\_splatting).

Table G.24 Attributes inherited from glTF and KHR\_gaussian\_splatting

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute semantic** | **Accessor type** | **Required** | **Description** |
| POSITION | VEC3 | Yes | Splat center position (glTF base attribute). |
| COLOR\_0 | VEC3 or VEC4 | Recommended | Baseline color for fallback point rendering. COLOR\_0.a should carry opacity when available. COLOR\_0.rgb is also used for degree 0 reconstruction when shEncoding.dcFromColor0 is true. |
| KHR\_gaussian\_splatting:ROTATION | VEC4 | Yes | Quaternion rotation (x,y,z,w) defining local axes orientation of each splat. |
| KHR\_gaussian\_splatting:SCALE | VEC3 | Yes | Per-axis scale in log-space as defined by KHR\_gaussian\_splatting. |
| KHR\_gaussian\_splatting:OPACITY | SCALAR | Yes | Opacity in the range [0,1] as defined by KHR\_gaussian\_splatting. |
| KHR\_gaussian\_splatting:SH\_DEGREE\_0\_COEF\_0 | VEC3 | Conditional | Degree 0 SH coefficients. Required when SH is used and shEncoding.dcFromColor0 is false. |
| KHR\_gaussian\_splatting:SH\_DEGREE\_1\_COEF\_n | VEC3 | Conditional | Degree 1 SH coefficients (n = 0..2). Present only when shEncoding.layout is khrPacked and maxDegree >= 1. |
| KHR\_gaussian\_splatting:SH\_DEGREE\_2\_COEF\_n | VEC3 | Conditional | Degree 2 SH coefficients (n = 0..4). Present only when shEncoding.layout is khrPacked and maxDegree >= 2. |
| KHR\_gaussian\_splatting:SH\_DEGREE\_3\_COEF\_n | VEC3 | Conditional | Degree 3 SH coefficients (n = 0..6). Present only when shEncoding.layout is khrPacked and maxDegree >= 3. |

The MPEG\_gaussian\_splatting\_transport extension defines the additional SH coefficient accessors used when MPEG SH layouts are selected.

Table G.25 gives the semantics of the MPEG\_gaussian\_splatting\_transport extension

Table G.25 — Semantics of the MPEG\_gaussian\_splatting\_transport extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| shEncoding | object | M |  | Signals how SH coefficients are encoded for this splat primitive. The semantics of this object are given in table G.26. |
| stitching | object | O | N/A | Contains parameters to attach a Gaussian Splat to a 3D triangular mesh primitive. The semantics of this object are given in table G.27 |

Table G.26 — Semantics of a MPEG\_gaussian\_splatting\_transport.shEncoding object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| layout | string | M |  | Indicates the encoding mode of the SH coefficients.  Possible value are: “mpegProgressive”, "mpegPerChannel" or "khrPacked".  If "khrPacked", SH coefficients are stored directly using the KHR\_gaussian\_splatting SH\_DEGREE\_l\_COEF\_n VEC3 attributes. |
| maxDegree | integer | M |  | Maximum SH degree available to the receiver. |
| dcFromColor0 | boolean | M |  | If true, receivers reconstruct the degree-0 (DC) term from COLOR\_0.rgb. If false, degree-0 is carried explicitly via KHR\_gaussian\_splatting:SH\_DEGREE\_0\_COEF\_0. |
| mpegProgressive { | Object | CM |  | If layout is "mpegProgressive", SH coefficients are stored using MPEG progressive degree-group accessors (SCALAR). FIRST covers degree 1, SECOND covers degree 2, THIRD covers degree 3. The DC term is not stored in these accessors. |
| first | integer | M |  | The index of the accessor that contains degree 1 coefficients (9 scalar values per splat) as 3 RGB triplets (coef 0..2). The DC term is excluded. |
| second | integer | CM |  | The index of the accessor that contains degree 2 coefficients (15 scalar values per splat) as 5 RGB triplets (coef 0..4). The DC term is excluded. |
| third | integer | O | N/A | The index of the accessor that contains degree 3 coefficients (21 scalar values per splat) as 7 RGB triplets (coef 0..6). The DC term is excluded. |
| } |  |  |  |  |
| mpegPerChannel { | Object | CM |  | If layout is "mpegPerChannel", SH coefficients are stored using MPEG per-channel accessors (SCALAR), one accessor each for R, G, and B. The DC term is not stored in these accessors. |
| r | integer | M |  | The index of the accessor that contains red channel coefficients (15 scalar values per splat) for degrees 1 to 3, in order: degree 1 (3), degree 2 (5), degree 3 (7). The DC term is excluded. |
| g | integer | M |  | The index of the accessor that contains green channel coefficients (15 scalar values per splat) for degrees 1 to 3, in order: degree 1 (3), degree 2 (5), degree 3 (7). The DC term is excluded. |
| b | integer | M |  | The index of the accessor that contains blue channel coefficients (15 scalar values per splat) for degrees 1 to 3, in order: degree 1 (3), degree 2 (5), degree 3 (7). The DC term is excluded. |
| } |  |  |  |  |
| progressive { | object | O | N/A | Provides progressive ordering of accessors. |
| stages | Array[stage] | M |  | Array of progressive stages. Stages are ordered from lower to higher fidelity. In its accessors array property, a stage object lists accessors that should be fetched together. The listed accessors may be static or timed. An accessor is timed if and only if it carries MPEG\_accessor\_timed. |
| } |  |  |  |  |

Table G.27 Semantics of a MPEG\_gaussian\_splatting\_transport.stitching object

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | | **Type** | **Usage** | **Default** | **Description** |
| mesh | | integer | CM | N/A | Index of the mesh in the glTF meshes array, to which the GS is stitched. |
| primitive | integer | | CM | N/A | Index of the mesh primitive to which the GS is stitched. |
| vertices | integer | | CM |  | The index of the accessor that contains the vertex indices. It defines the location of a Gaussian Splat. If present “stitching.faces” and “stitching.weights” shall not be present. |
| faces | integer | | CM |  | The index of the accessor that contains the face indices. It defines the location of a Gaussian Splat in a triangular face. If present “stitching.vertices” shall not be present and “stitching.weights” are mandatory. |
| weights | integer | | O | N/A | The index of the accessor that contains the barycenter weights. It provides a triplet (VEC3) that are the barycenter weights to define the location of a Gaussian Splat in a triangular face. |
| distances | integer | | O | N/A | The index of the accessor that contains the scalar distances, i.e. the scalar distance from the mesh for a Gaussian Splat. The distance shall be applied along the normal of the attribute present e.g., a vertex or a face. |
| displacement | | integer | O | N/A | The index of the accessor that contains the displacement vector (VEC3), i.e. the vector displacement distance from the mesh for a Gaussian Splat. |

Note: In the MPEG SH layouts, the DC (degree 0) term is not stored in the MPEG coefficient accessors. When shEncoding.dcFromColor0 is true, the receiver reconstructs degree 0 from COLOR\_0.rgb. When shEncoding.dcFromColor0 is false, degree 0 is carried explicitly using KHR\_gaussian\_splatting:SH\_DEGREE\_0\_COEF\_0.

The MPEG\_gaussian\_splatting\_transport extension adds transport-level signaling while keeping KHR\_gaussian\_splatting as the semantic baseline. It shall be carried inside the KHR\_gaussian\_splatting.extensions.

It provides:

* accessor bindings for the MPEG SH layouts and DC reconstruction policy, and
* an optional progressive ordering over accessors. Any referenced accessor is treated as timed if and only if it carries MPEG\_accessor\_timed.
* An optional stitching mechanism, that permits to attach a Gaussian Splat to a 3D triangular mesh primitive. The 3D mesh and its primitive are explicitly signaled in the stitching object if present. The stitching object is compatible with all modes.

### G.5.3 Processing Model

Three SH encoding modes are supported:

* khrPacked: SH coefficients are stored directly using the KHR\_gaussian\_splatting:SH\_DEGREE\_l\_COEF\_n VEC3 attributes.
* mpegProgressive: SH coefficients are stored in three SCALAR arrays corresponding to degree groups. Per splat, FIRST contains 9 values (3 RGB coefficients for degree 1), SECOND contains 15 values (5 RGB coefficients for degree 2), and THIRD contains 21 values (7 RGB coefficients for degree 3).
* mpegPerChannel: SH coefficients are stored in three SCALAR arrays, one per channel R, G, and B. Per splat, each channel array contains 15 values in the order degree 1 (3), degree 2 (5), degree 3 (7).

In both MPEG layouts, the DC term shall not be included in these arrays. When dcFromColor0 is true, the receiver shall reconstruct the degree 0 coefficients from COLOR\_0.rgb using rules consistent with the declared KHR\_gaussian\_splatting colorSpace. When dcFromColor0 is false, degree 0 coefficients are carried explicitly using KHR\_gaussian\_splatting:SH\_DEGREE\_0\_COEF\_0.

Progressive download is signalled by listing accessors in progressive.stages, in increasing order of fidelity. A receiver may initially fetch only the first stage and render using whatever attributes are available. As additional stages become available, the receiver progressively refines the rendering without re-decoding previously available data.

To avoid partial states, a stage should correspond to complete attributes or complete SH degrees. For example, when shDegree2 becomes available, all 5 degree 2 coefficient vectors are available for every splat.

For dynamic sequences, timed delivery is supported by using MPEG timed media mechanisms. An accessor referenced by the primitive attributes or by MPEG\_gaussian\_splatting\_transport is treated as time-varying if and only if the accessor carries the MPEG\_accessor\_timed extension. Timed accessors are backed by circular buffers as defined by MPEG-I Scene Description.

At each rendering cycle, the receiver uses the newest complete set of per-frame attributes available in the circular buffer. If no new data is available, the receiver continues using the last available set.

For any mode the object stitching can be present, please note that this is optional. If the stitching object is present the attributes “mesh” and “primitive” are conditional mandatory.

The attribute “mesh” references a mesh object in the list of mesh objects. If “mesh” is present and “primitive” is not present, the reference mesh shall have a single primitive, and this primitive contains the 3D mesh to stitched the Gaussian Splats. If “mesh” is present and primitive is present the Gaussian Splats are attached to the referenced “mesh[id].primitive[id]” given by the two attributes.

The attribute “primitive” references a primitive object of a mesh object. If “mesh” is not present, the “primitive” references a primitive within the mesh object that the extension KHR\_gaussian\_splatting is included. If “mesh” is present, the “primitive” references a primitive of the referenced “mesh”.

When the attribute “vertices” is present the attributes “faces” and “weights” are not present, and vice versa.

*Annex G*

Add the following new Clause as follows:

**G.6 Support for stereo applications**

**G.6.1 Introduction**

To integrate the support of stereo codec applications in MPEG Scene Description, a glTF extension, identified by MPEG\_Material\_stereo, is used.

Each mesh primitive has a reference to a material that defines how the surface of a 3D object looks when rendered. The MPEG\_material\_stereo extension is applied to the material property. The mesh defines the geometry of the object, while the MPEG\_material\_stereo extension aggregates all relevant texture information processed for rendering.

**G.6.2 Semantics**

The semantics of the MPEG\_material\_stereo extension is shown in Table G.28.

**Table G.28: Semantics of the MPEG\_material\_stereo extension**

| **Name** | **Type** | **Default** | **Usage** | **Description** |
| --- | --- | --- | --- | --- |
| layers | array[Layer] | N/A | M | Provides an array of layer objects, limited to 2 layers. See Table G.27 for the semantics of a layer object. |
| **Legend:**  For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory. | | | | |

In Table G.29, a layer is an object which combines a texture with associated rendering properties. Each texture is described in an item of the *layers* property of the MPEG\_material\_stereo extension. Additional properties in the layer object indicate whether the texture corresponds to a primary reference layer or a secondary layer and specify its association with the left or right eye.

**Table G.29: Semantics of a MPEG\_material\_stereo layer object**

| **Name** | **Type** | **Default** | **Usage** | **Description** |
| --- | --- | --- | --- | --- |
| primaryReference | boolean | N/A | M | Indicates whether this layer component is the primary component (also known as the hero view) among other layers. When primaryComponent is set to True, this indicates the layer component is a primary component.  Only one layer among the layers of the media object shall have the primaryReference set to True. |
| texture | integer | N/A | M | Provides the index of a texture. |
| eyeChannel | integer | N/A | M | Indicates if the information refers to left (0) or right (1) eye . |
| **Legend:**  For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory. | | | | |

**G.6.3 Processing Model**

Each mesh with a reference to a material with a MPEG\_material\_stereo extension, references a material that aggregates stereo layers. Within the extension, exactly one layer shall be identified as the primary reference, while the additional layer may be associated with the left or right eye. A conforming implementation should combine these layers according to their designated eye channel, ensuring that the primary reference is consistently used as the basis of rendering. In case when stereo rendering is not supported by the Presentation Engine (PE), one texture which is identified as the primary reference layer in the MPEG\_material\_stereo extension is used. When the MPEG\_material\_stereo extension is not supported, the standard material glTF element can be used as fallback.

The textures may be linked to glTF images or to items in the media array in MPEG\_media extension. In this latter case, the Media Access Function (MAF) decodes, and processes related media item(s) to produce reference(s) to corresponding texture(s). The MAF may use *ViewInfo API* to process and generate the view(s) according to the user’s pose information.

A glTF file with stereo media content indicates MPEG\_material\_stereo extension in the extensionsUsed property.

When a scene description file becomes available, the PE shall proceed as follows:

1. Parse the related glTF file
2. Extract the scenes, nodes and meshes arrays
3. For each mesh referencing a material with a MPEG\_material\_stereo extension, process the extension data to aggregate stereo layers information. For each layer, extract:
   1. The primary reference flag (primaryReference) indicates whether this layer is designated as the primary reference.
   2. The index of the texture (texture) refers to the index of the texture within the glTF textures array.
   3. The eyeChannel information (eyeChannel) indicates the association of the layer with the left eye or the right eye.
4. According to the device stereo capability,
   1. If stereo rendering is possible, for each stereo layer identified in Step 3, load the texture referenced by its texture index from the glTF textures array
   2. For non-stereo device, load the texture which is identified as the primary reference by its texture index from the glTF textures array
5. Once all texture(s) is(are) loaded, the PE applies the material to the mesh primitive according to PE rendering capabilities.
   1. If stereo rendering is possible, the rendering is done for each eye. If a stereo layer is missing for one of the eyes, the rendering for that eye falls back to using only the primary reference layer.
   2. For non-stereo device, the rendering is done using the primary reference layer.

*Add the following Annex I:*

1. (normative)  
     
   Support for multi-users interactivity
   1. General

This annex is about multi-users application where multiple users meet in a shared space. This space may be described with MPEG-SD and may contain interactivity features as specified in section 8.2.

The triggers, as specified in section 8.2.2.1, do not fully support interactions that involve multiple users.

For the TRIGGER\_USER\_INPUT trigger, the trigger is evaluated for each user and the actions associated with the trigger may be performed multiple times, for each received user’s input that met the trigger condition.

For a TRIGGER\_USER\_INPUT trigger involving multiple users, the trigger must be evaluated for all the users and the actions associated with the trigger must be performed only once, when all the user’s inputs met the trigger condition (or when at least one depending on the scenario).

This Annex provides semantics that address multi-users triggers.

* 1. Syntax

The following table give the semantics of a trigger object, as specified in the table 33 of the section 8.2.2.1, with a new “multiUsers” parameters added in the *extras* field.

Table 33 — semantics of a trigger

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| type | enumeration | M |  | One element that defines the type of the trigger. |
| … |  |  |  |  |
| extras | object | O | N/A |  |
| multiUsers | object | O | N/A | Additional parameters to address a TRIGGER\_USER\_INPUT trigger involving multiple users. See Table I.2- 1 for the details of this object. |

Table I.2 — 1: semantics of a multiUsers object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Usage** | **Default** | **Description** |
| multiCondition | enumeration | O | ALL | If ALL, the trigger’s condition must be met by all the users.    If ONLYN, the trigger’s condition must be met by exactly N users.    If ATLEASTN, the trigger’s condition must be met by at least N users. |
| duration | number | O | 2 | Give a time duration, in second.    After the trigger condition is met by a first user, it specifies a time window during which the *multiCondition* must be met. |
| number | number | O | 1 | Give the number of users when condition is ONLYN or ATLEASTN. |

* 1. Processing model

After receiving a scene description file, the application parses the file and starts collecting information from all the connected users (user input, pose…).

The application then enters the rendering loop which include a scene update phase performed by the scene manager (SM), that processes the scene interactivity by iterating over the behaviors and manage the other elements of the scene description (animation, lighting, physics…).

The iteration over the behaviors is detailed here after, based on the diagram in the Figure I.3- 1.

During the processing of the scene interactivity, for each behavior and for each trigger referenced in a behavior (1):

— (2) The SM checks if the trigger involves multiple users i.e. a *multiUsers* parameter is specified in the triggers description.

— (3 to 5) If the trigger is not multi-users, for each user information that meet the trigger’s condition, the SM executes the actions referenced in the behavior. It then continues the processing with the next behavior (1).

— (6) If the trigger is multi-users, the SM check if the trigger condition is met for at least one user:

— if yes, (7) the SM checks if the condition specified in the *multiUsers* object is met, i.e. the *multiCondition* has been met during the given *duration*:

— If yes, the SM executes the actions referenced in the behavior (8).

— if no, the processing continues with the next behavior (1).

— if no, the processing continues with the next behavior (1).

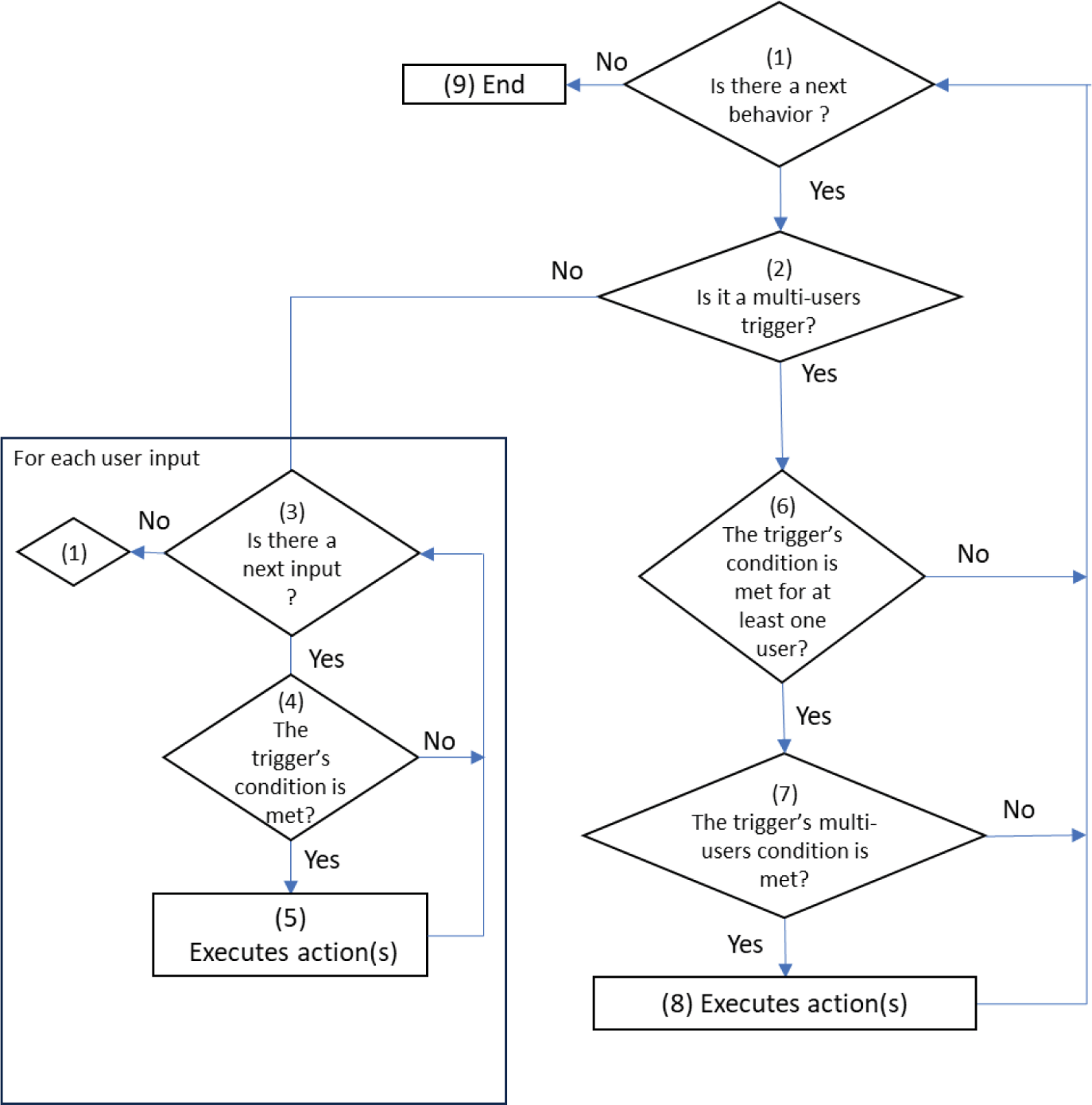


Figure I.1 — MPEG-SD triggers processing

*Bibliography*

Add the following reference to the bibliography

[10]   “What is the Coulomb friction law”: <https://www.encyclopedie-environnement.org/en/zoom/what-is-the-coulomb-friction-law/>