A black and white logo

Description automatically generated ISO/IEC JTC 1/SC 29/WG 3 N01050

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

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

**Document type:** Output Document

**Title:** Requirements coverage for MPEG-I Scene description

**Status:** Approved

**Date of document:** 2023-10-24

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

# Expected action: ACT

**Action due date:** None .

**No. of pages:** 21 (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 N01050**

|  |  |
| --- | --- |
| **Title** | **Requirements Coverage of MPEG-I Scene Description** |
| **Source** | **Systems** |
| **Status** | **Approved** |
| **MDS** | **230203** |

**Introduction**

This document provides the status of the MPEG-I requirement coverage identified for scene description as achieved in SD first edition, first and second amendments.

The following table provides these columns:

* Column 1: Requirements numbers as documented in MPEG 23246.
* Column 2: Requirements.
* Column 3: Fulfillment by gltf2.0 or other specifications: Does gltf2.0 or other specifications fulfill this and how?
  + Pink: completely fulfilled by glTF2.0, or other specification, implementation aspects and no MPEG SD work necessary.
  + Yellow: partially fulfilled by glTF2.0, or other specification but needs MPEG extensions.
  + Orange: partially fulfilled by glTF2.0, but prefer/ask Khronos to do extension
  + Blue: not fulfilled by glTF2.0, or other spec needs MPEG SD extension
* Column 4: Status MPEG Extensions: documents the basic principles for the MPEG extensions and the status in the latest DIS (SD version 1)
* Column 5: The overall status of the requirement as of Oct 2023 – When the requirement is fulfilled, either by SD (version 1, and amd1&2) or in other Spec (e.g. gltf, haptic, audio etc), it is indicated by a Green color coding in that column. When not fully completed some comments are also included (e.g. proposals: other spec, later phase of SD).
* Column 6: Suitability as Test Scenario: Documents how this requirement could be demonstrated.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Number | Requirements N19511 and N00230 | | Fulfillment gltf2.0, including N18869 | Status MPEG Extensions from N00369 | Overall status MPEG #143 | Suitability as Test Scenario |
|  | | Phase 2a Requirements | | | | |
| General | | | |  |  | 1 – must have  2 – quite important  3 – nice to have  4 – if there is lots of time  5 – unnecessary |
| 4 | If possible, the solution shall define interfaces to integrate existing scene description formats rather than define a new scene description format. | | Gltf2.0 is an existing scene description. The design based on glTF can be used as a blueprint for developing similar extensions to other existing scene description formats such as USD and OSG. |  | Completed | 1 basic test. Use gltf. |
| 5 | The solution shall reuse existing interfaces/API definitions (also from other SDOs) whenever possible and appropriate | | Gltf2.0 basic functionalities to build APIs on top of it. Relevant ones will be defined by MPEG but re-using also concepts defined in W3C for HTML-5. | Addressed in DIS, APIs between presentation engine and media access functions defined. | Partially completed in SD. Evaluate API to audio renderer, possibly others API  Proposed for phase 3 | 1 basic test. Use gltf. |
| 6 | It shall be possible to indicate object information in the scene description format such that one can derive a consistent relationship between object map information in a video bitstream and object information signaled in the scene description format | | This requirement is primarily addressed by the media access function.  In addition, information in the scene description takes precedence over information in the media and shall always be present. The scene description may refer to object information in the stream, but in this case, it requires that the information if provided through a well-defined buffer to the presentation engine. | Addressed in DIS as part of the media access function and the buffer management. | Completed SD 1st edition | 1 add a V-PCC/MIV object to a gltf scene.  Add a 360 object to a gltf scene  Combine a V-PCC/MIV object with a 360 object. |
| 7 | The information signaled in the scene description format shall be consistent with presentation layer information signaled in the video/audio bitstream at the codec layer as well as information signaled in the media formats at the systems level via ISOBMFF, DASH, etc… In case of contradicting information, the scene description format shall take precedence | | This needs to be well-documented in the standard and requires detailed and consistent information for each MPEG media that is added to Scene description. It is a matter of well-defined extensions.  The MPEG-I architecture has been designed to properly decouple the media format as it is stored or delivered from the actual media data that is used for rendering. The Presentation Engine expects media data to be in raw graphics/audio formats (e.g. vertex buffers, color buffers, normal buffers, audio sample buffers, …). Any decoding and processing of the actual media to reformat it for rendering is done in the processing pipeline and is guided by the Media Access Function based on information received from the scene graph.  The following figure shows how the pipelines can be constructed to feed raw data into the buffers, which will be used for rendering by the Presentation Engine.  A close up of a screen  Description automatically generated | Addressed in DIS as part of the media access function and the buffer management. | Completed SD 1st edition | 2  See above under 6 |
| 8 | It shall be possible to associate Information in the media bitstream targeted for the scene description with a processing model. | | Based on the reference design, it requires that metadata in the bitstream is loaded into buffers for consumption the presentation engine with well-defined syntax and semantics.  Buffers may be static or dynamic. | Under development. Basic functions defined in DIS, but needs more work especially on integrating MPEG-defined media formats such as OMAF, PCC, etc.  Also dynamic metadata, for example carried in SEI messages such as recommended viewports would need processing models.  If processing model is not provided, such metadata is expected to be “lost”/”ignored” in the MAF processing. | Completed for phase 2 scenarios, phase 3 will consider additional use case as needed. | 1  See test scenarios under 6. |
| 9 | The scene description shall enable the option to describe the scene using geographical coordinate systems. | | glTF2 has currently no support for geographical coordinate systems. | Under development in TUC as part of the media access function and the buffer management. | Completed in AMD2 with AR anchor and geospatial trackable | 3  Test scenario tbd. |
| 10 | The scene description shall enable modular rendering, i.e. smaller portions of the scene can be independently accessed and rendered. | | The current design permits to only access media that is in the viewport of the scene. The MAF design allows to optimize the access to data according to the current user pose and viewport. | Addressed in the in the DIS as part of the media access function and the buffer management. | Completed in SD ed1 | 3  Test scenario tbd, but requires multiple objects at different viewports. For example two room scenario, where you can walk through. |
| Reference Scene Description Selection | | | |  |  |  |
| 11 | The scene description shall support audio, video and other media formats standardised by MPEG. | | glTF2 has currently no support for audio or video media formats. However, glTF supports several still image formats. | Partially addressed in DIS. Different approaches taken   * Use glTF to describe 2D buffers. Buffers are circular * Audio extensions also added.   Additional information in TUC and needs more work, for example to directly integrate 3D objects such as PCC or 360. | Mostly completed in first edition (AV) and Amd1/Amd2, PCC, MIV, haptic v1)  Additional features may be required for Haptics phase 2 and later MPEG codec (e.g. VD-Mesh); with support in SD phase 3. | 1  For each of the three approaches a test scenario is desired.  Some overlap with the one in 6  Definitely need an A/V scenario. |
| 12 | The scene description shall enable the support of other visual or audio media formats. | | With addressing 2b, media other than MPEG can be supported by adding such media to MPEG containers.  By setting the up the MAF pipelines, one can also use other codecs. As long as the raw format is supported, this is independent of the codecs.  Glft itself supports several 3D image formats (Alembic, etc.) | Mostly addressed in the DIS.  Some improvements on terminology expected, e.g.  *Raw* format: (2D, 360, Point cloud, mesh, etc.,)  *Coded* format: (compressed 2D or raw format, e.g. V-PCC, G-PCC, Draco, Alembic, OpenVDB, …)  *Track* formats: time-synced and packaged for delivery, accessed by MAF (MP4, CMAF, etc.) | completed | 3  Combining MPEG media with for example Draco may be a nice test scenario. |
| 13 | The scene description shall support definitions to indicate how sub-graphs and objects are related in terms of their temporal, spatial and logical relationships | | This requirement is partially supported, since with exception of animations, all nodes of a scene graph are assumed to be active at time 0 and there is no concept of scene updates in glTF2. Regarding the logical relationships between media elements within a scene, means of interactivity and possible constrains to it should be defined. | Partially supported in gltf by nodes (spatially and logically).  Scene description updates are under development and partially addressed in DIS. More work in EEs and TuC. | Partially Mostly completed. Event/interactivity based update to be done in phase 3. Some technologies in the TuC | 4 |
| 14 | The scene description shall support composition of digital representations of natural and synthetic objects. | | Gltf permits to add different objects through nodes and buffers into a scene representation. Objects can be then be rendered by a gltf renderer. |  | Completed. If the requirement is extended to AR scenario, it will be addressed in phase 3. | 1  Likely covered by one of the above scenarios. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 15 | The scene description shall support synchronisation between objects and attributes in the scene. | This is only supported through animations at the moment. MPEG needs to provide appropriate extensions to address this issue. | Addressed in the DIS. Timed animation is supported | Completed in SD 1st edition | 1  Likely covered by one of the above scenarios. |
| 16 | support temporal random access. | glTF supports spatial random access through node hierarchy, | Under development in DIS, but more work is carried in EEs. | Completed with SD support for temporal RA | 2 - 3  May be an extension to one of the above scenarios. |
| 16.1 | support spatial random access. | MPEG needs to provide appropriate extensions to address this issue. |  | Spatial RA in phase 3. (LOD or viewport or spatial partitioning) |  |
| 17 | The scene description should support information to enable a renderer to perform path tracing. | glTF2 supports path tracing extensions.  Physically Based Rendering is collectively used to denote a set of rendering techniques that aim at providing an accurate rendering of the scene that is indistinguishable from a photograph of the real scene. These techniques usually rely on precise calculations of the light transfer in the scene by estimating the reflection and refraction of light, coming from all light sources in the scene, on the objects available on the scene. For these techniques to work properly, there is need to properly describe the materials of the objects and how these materials interact with light as well as the ability to describe the geometry orientation of the objects themselves. In addition, there is need to describe light sources in the scene.  Physically Based Rendering material feature originally came from an extension, but it was made part of the GLTF 2 standard. Values are stored in a material asset that gets applied to the GLTF file's geometry; some values vary per pixel of a texture map, and some are constants that get used equally everywhere the material is assigned.  As for the shader that then interprets the material values, there are GLTF extensions that allow custom user-created shaders, but that's not part of a standard GLTF 2 workflow. Normally, the PBR shader that interprets those material values is outside the GLTF file, supplied by the app reading it. The idea of the PBR approach is to standardize one flexible "ubershader" implementation across many different apps--so the final output should look fairly similar everywhere.  The core functionalities are here:   * pbrMetallicRoughness material model * normalTexture, emissiveTexture, and occlusionTexture map support for these materials * A non-normative BRDF implementation   The extensions are here:   * [KHR\_materials\_clearcoat](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_materials_clearcoat/README.md) * [KHR\_materials\_pbrSpecularGlossiness](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_materials_pbrSpecularGlossiness/README.md) * [KHR\_materials\_unlit](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_materials_unlit/README.md) * [KHR\_lights\_punctual](https://github.com/KhronosGroup/glTF/blob/master/extensions/2.0/Khronos/KHR_lights_punctual/README.md)   In addition, the following vendor specific extensions are also available:   * EXT\_lights\_image\_based which stores light probes from the scene in an environment map * ADOBE\_materials\_thin\_transparency to represent materials with transparency   Ray tracing is one of the techniques to achieve PBR. It is usually implemented in render passes that iterate over each pixel and cast light rays ending in that pixel to determine which objects will be reflected on that pixel by performing a hit test on the scene geometry. The way the rays reflect on the surfaces of these objects is determined by the object’s material characteristics. The above tools and extensions allow for an accurate implementation of ray tracing on a scene described by glTF. |  | Completed | 3 |
| 18 | The scene description shall support sub-graph representation that allows modular rendering e.g. leafs in the scene description tree can also be packaged and referenced individually from a parent scene description and container. | glTF2 supports modular design of leaves by buffers.  The following diagram shows the structure of glTF 2.0:  A picture containing electronics, control  Description automatically generated  It starts with a **scene** node which in turns has a hierarchy of nodes (a set of disjoint trees). This is exactly the definition of a scene graph, albeit with a limitation of nodes having a single parent for simple parsing and rendering.  The example assets that are provided show how a user can navigate the scenes, e.g. walking inside a conference room or living room, where other 3D assets/objects may also be placed. If in doubt, effort should be done to define **test scenarios** to validate capabilities of the technology. |  | Completed | 3 |
| 19 | The scene description shall support references (e.g. URLs) to external media resources in place of embedded file references. | glTF2 supports external media by reference.  Extensions for MPEG are needed to address different delivery use cases. | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | Completed SD 1st edition and AMD1 | 1  Likely covered by one or several of the above scenarios |
| 20 | The scene description shall support a mechanism to safely customize behavior for nodes like camera, texture, geometry, audio, and object placement nodes through sandboxed, validated domain specific shaders or scripts for these nodes without affecting the functionality or forcing changes to the root node graph or other node types; i.e. provide a mechanism to safely extend the scene description. | Gltf2.0 support this. The following statements can be found in the glTF 2.0 specification:  *The GL Transmission Format (glTF) is an API-neutral runtime asset delivery format. glTF bridges the gap between 3D content creation tools and modern 3D applications by providing an efficient,* ***extensible****, interoperable format for the transmission and loading of 3D content.*  *glTF solves these problems by providing a vendor- and runtime-neutral format that can be loaded and rendered with minimal processing. The format combines an easily parseable* ***JSON scene description*** *with one or more binary files representing geometry, animations, and other rich data.*  *Extensibility. While the initial base specification supports a rich feature set, there will be many opportunities for growth and improvement. glTF defines a mechanism that allows the addition of both general-purpose and* ***vendor-specific*** *extensions.* |  | Completed | 4 |
| I-s Interface: Presentation engine interface | | |  |  |  |
| 21 | It shall be possible to update the whole scene-graph, a sub-graph, or a node in the scene description | glTF2 does not come with a scene update mechanism.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | Partially completed. Event based update to be done in phase 3 | 1  Need to define a scenario. This is part of the current discussion |
| 22 | It shall be possible to correctly render a 6DoF Presentation after a random access in time | glTF2 does neither have support for a timing model nor scene updates through time and as such every glTF2 is considered a random access point in time.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | Completed with support for temporal RA | 2  Likely covered by extension from above |
| 23 | It shall be possible to perform timed scene description updates | glTF2 does not come with a scene update mechanism.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | Completed with scene update extension | 1  See 21 |
| 24 | It shall be possible to associate a scene description update with the corresponding scene description | glTF2 does not come with a scene update mechanism.  MPEG needs to provide appropriate extensions to address this issue | Partially supported in the DIS, work ongoing in EEs. | Completed with scene update extension | 2  See 21 |
| 25 | It shall be possible to use a scene description as the entry point to a 6DoF presentation | Gltf2.0 supports exactly this. |  | Completed | 1  Likely covered by many test scenarios |
| I-m Interface: Media access Interface | | |  |  |  |
| 26 | It shall be possible to access timed and non-timed, 2D and 3D media (meshes, point clouds, audio elements, …), stored locally or over the network | glTF2 has support for buffers and images that fetch their content from a local file system or over the network. However, there is no support for timed media.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | Completed | 1  Likely covered by many test scenarios |
| 27 | It shall be possible to pre-fetch media that the presentation engine expects to be used in the presentation | glTF2 has support for buffers and images that fetch their content from a local file system or over the network. However, there is no support for timed media.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | Completed | 3  Nice to have, but not relevant. We know that MPEG delivery can fulfill this. |
| 28 | It shall be possible to retrieve media depending on the desired level of detail | Retrieving different levels of detail of geometry and texture is generally possible in glTF2 through pre-processing the glTF2 scene. These operations are not supported by the glTF2 scene itself.  MPEG needs to provide appropriate extensions to address this issue | Partially addressed in DIS using MAF and APIs from GLTF to MPEG media delivery (for example your viewing position).  More work may be needed. | Completed | 3  Would be nice, but requires likely some work. |
| 29 | It shall be possible to retrieve and access referenced media partially in time and space | Currently, glTF2 references lack any attributes to indicate time or space points in the referenced media. Several MPEG media components offer partial access capabilities, as such means to describe such partial access in time and space should be defined.  MPEG needs to provide appropriate extensions to address this issue | Partially addressed in DIS using MAF and APIs from GLTF to MPEG media delivery (for example your viewing position). This allows to use viewport-dependent streaming as provided by MPEG media-  More work may be needed. | Completed | 3  Would be nice, but requires likely some work. |
| 30 | It shall be possible to describe position, orientation, and visual/acoustic characteristics when rendering referenced media | Supported for visual objects through gltf2.0.  glTF2 has no support for audio nodes.  MPEG needs to provide appropriate extensions to address this issue  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS for audio extensions. | Completed with audio extension and MPEG\_texture\_video attached to a node material | 1  Covered by above scenarios |
| 31 | It shall be possible to synchronize media objects/resources and media components of a single object | glTF2 has no support for timed media nodes.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using MAF and APIs from GLTF to MPEG media delivery. | Completed | 1  Covered by above scenarios |
| 32 | Audio elements shall be rendered consistently with their corresponding visual elements, if such visual elements exist. | glTF2 has no support for audio nodes.  MPEG needs to provide appropriate extensions to address this issue | Addressed in DIS using, audio extensions, MAF and circular buffers. | Completed for audio. Phase 3 for immersive audio | 1  Covered by above scenarios |
| 33 | The specification shall enable synchronization of audio and video of users and the scene. | glTF2 has no support for audio from the scene or from the user.  MPEG needs to provide appropriate extensions to address this issue  May also relate to real-time | Addressed in DIS using, audio extensions, MAF and circular buffers. Real-time support for users is likely not be part of phase 2a. | Completed for audio.  Phase 3 for Immersive audio | 1  Covered by above scenarios |
| Delivery Requirements for Content Interactivity | | |  |  |  |
| 38 | The specification shall support low delay processing and presentation of object features for a scene, in order to minimize motion-to-photon latency. |  |  | Enabled by SD Addressed at the implementation level  . |  |
| 39 | The specification shall support defining conditional switching between viewports |  |  | Partially supported by multiple camera. May need some SD support in Phase 3 |  |
| 40 | The specification shall support hotspots that trigger actions like switching viewports. |  |  | Completed. Supported in SD with the proximity trigger of the interactivity extension |  |
| 41 | The specification shall support signalling how content needs to loop back or continue playing, where this behaviour may be triggered by certain interactive conditions. |  |  | Completed. Supported in SD with the proximity trigger of the interactivity extension |  |
| Phase 2b | | | | | |
| Immersive Audio | | | | | |
| 48 to 74 |  |  |  | Partially Completed. Addressed in current and future audio specification |  |
| Reference Scene Description Selection | | | | | |
| 77 | The scene description should support nodes and attributes in order to implement natural laws of light, energy propagation and physical kinematic operations. | . |  | SD: support of physics parameter in the interactivity extension. |  |
| 78 | The scene description should support nodes and attributes in order to implement natural laws of acoustic energy propagation and physical kinematic operations. | The support for acoustic characteristics of materials and nodes is non-existent in glTF2.  MPEG needs to provide appropriate extensions to address this issue. | Basic principles addressed for audio extensions in DIS  More advanced features are part of MPEG-I audio available for phase 2b. | Completed for Audio.  Phase 3 for immersive audio |  |
| 79 | The scene description should support description of ray-traced camera parameters for rendering | glTF2 supports ray tracing extensions. See requirement 17. |  | Completed |  |
| 80 | The scene description shall support parametric models for use in rendering environmental acoustic behaviour (e.g. reverberation, occlusion and directivity). | glTF2 has no support for audio in the scene.  MPEG needs to provide appropriate extensions to address this issue. | Basic principles addressed for audio extensions in DIS  More advanced features are part of MPEG-I audio available for phase 2b. | Partially completed.  Immersive audio and Haptic phase 2 in SD phase 3 |  |
| I-l Interface: Local capture Interface | | |  |  |  |
| 81 | It shall be possible to discover and configure local capture modalities | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work | SD Phase 3 for haptic support in particular |  |
| 82 | It shall be possible to adjust the presentation based on local capture modality availability | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work | Phase 3 |  |
| 83 | It shall be possible to reference media objects that are captured locally using different capture modalities | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work | Phase 3 |  |
| 84 | It shall be possible to provide feedback through available actuators | glTF2 itself has no support for local modalities but OpenXR provides APIs to do that.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work | Should partially be covered by haptic phase 2. Need support in phase 3 of SD. |  |
| I-i Interface: User inputs Interface | | |  |  |  |
| 85 | It shall be possible to discover user interactivity modules | glTF2.0 supports this, but interaction with timed media still needs to be defined.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work | Partially completed with Interactivity extension and OpenXR.  SD Phase 3 for haptics and avatar |  |
| 86 | it shall be possible to define custom interactivity procedures based on input from the user or from the user’s devices and sensors | glTF2.0 supports this, but interaction with timed media still needs to be defined.  MPEG needs to provide appropriate extensions to address this issue | Left to phase 2b. Needs work | Partially completed with Interactivity extension and OpenXR.  SD Phase 3 in particular for haptics and avatar |  |
| Export | | |  |  |  |
| 87 | The scene description shall support information to enable a renderer to output raster data (image, and video), volumetric data (point clouds, meshes, arrays of voxels, and reflectance fields) and audio. | glTF 2.0 supports this requirement for all mentioned visual formats. An extension is required to extend the support or audio. | Audio support in DIS.  Raster data support in DIS  Volumetric data partially supported in DIS, more work is needed. | Partially completed in DIS, AMD1 and AMD2.  Phase 3 for immersive audio, Meshes |  |
| 88 | The scene description shall support a scriptable export output node for asynchronously exporting (as a file stream or buffer) any or all parts of the scene description connected to a node into a simpler or flattened representation. | glTF 2.0 is already a simple flat graph representation. Tools and scripts are available to convert it into a binary representation. |  | Completed. |  |
| Content Interactivity | | | |  |  |
| 89 | The specification shall enable realistic composition of a 6DoF scene depending on the user-selected location and orientation.  Note: Such composition may, e.g., include delivering proper lighting information and some form of geometry information of the scene so the view is rendered with realistic lighting and shadows. | Basic feature of a game engine.  SD as an enabler |  | Completed.  . |  |
| 90 | The specification shall support a user interacting with objects within a given virtual environment. |  |  | Completed with MPEG SD interactivity extension in AMD2 |  |
| 90.1 | The specification shall support giving an object a place in the virtual environment, moving it and changing some attributes  Example: a rectangular screen that represents a TV set on which a different channel could be chosen; a point cloud object that is inserted into a representation of the real world |  |  | Completed with MPEG SD interactivity, anchoring extensions and AMD1 |  |
| 91 | The specification shall support metadata describing objects and their features within a 6DoF scene. |  |  | Completed with AMD1 for V3C codec |  |
| 91.1 | This metadata will be defined in updated versions. | See MPEG SD TuC for semantic metadata |  | Phase 3 |  |
| 92 | The specification shall support replacement of 6DoF content media data. |  |  | Completed in interactivity extension if "replacement" means setting a spatial transformation of a graph node. |  |
| 92.1 | "The specification shall support metadata for the processing of object features, namely preserving and replacing features for objects in a scene, either by the client, or in the network. | Network aspect may be a phase 3 in SD or a different specification |  | Completed in SD.  . |  |
| Multi-User Interactivity | | | |  |  |
| 93 | The specification shall enable realistic composition of user-embodiment within 6DoF content media.  Note: Such composition may, e.g., include delivering proper lighting information and some form of geometry information of the scene so user-embodiment is rendered with realistic lighting and shadows. |  |  | Partial support with Avatar extension,  Additional support likely needed in phase 3 |  |
| 94 | The specification shall support rendering of other users in 6DoF content media, including possible speech or audio from other users. |  |  | Partial support with Avatar extension  Additional support likely needed in phase 3 |  |
| 95 | The specification shall enable multi-user immersive applications in which several users are experiencing the same immersive experience together. |  |  | Not completed yet, To be addressed in phase 3 |  |
| 95.1 | It shall be possible to detect & render interactions between users within the immersive environment. |  |  | Not completed yet, To be addressed in phase 3 |  |
| 96 | The specification shall support conveying metadata about the spatial alignment of a camera and the person that this camera is capturing, in the physical environment. This metadata shall include at least: | Gltf camera semantic as the basis for this |  |  |  |
| 96.1 | the distance between the camera and the captured person  Note: same requirement than 114 | Can be calculated by the Presentation Engine based on spatial attributes of the scene graph |  | Completed |  |
| 96.2 | the orientation of the camera |  |  | Completed: camera properties can be retrieved via maf api |  |
| 96.3 | the focal length and possibly other lens parameters |  |  | Completed: camera intrinsic can be retrieved via maf api |  |
| 96.4 | the location of the captured person’s head in the video | This requirement needs to be clarified. |  | phase 3. |  |
| Delivery Requirements for Multi-User Interactivity | | | |  |  |
| 97 | The specification shall enable multi-user immersive applications in which several users are experiencing the same immersive experience together. | See Technology under Consideration and agreed use cases. |  | Some support in SD needed. Other system spec TBD.  Not completed yet, To be addressed in phase 3, |  |
| 97.1 | " The specification shall enable synchronous play-out between multiple users consuming the same scene, where the synchronization is accurate within 100 ms.  Example: multiple users viewing the same sports event and communicating about the event in real-time  Note: the end-to-end latency of the experience is much less important than the inter-user sync. " |  |  | Some support in SD needed. Other system spec TBD.  Not completed yet if the same/shared scene can be modified by the user  Phase 3 |  |
| 98 | The specification shall support synchronization of audio and video of users and the scene. | see TuC and agreed use-cases |  | Some support in SD needed. Other system spec TBD.  Not completed yet, To be addressed in phase 3, |  |
| 98.1 | Synchronization among users of all interactions (with VR objects and between users) shall be supported. | see TuC and agreed use-cases |  | Some support in SD needed. Other system spec TBD.  Not completed yet, To be addressed in phase 3 |  |
| 99 | The specification shall support the low-latency delivery of interactions between users within a given virtual environment. | see TuC and agreed use-cases |  | Some support in SD needed. Other system spec TBD.  Not completed yet, To be addressed in phase 3. , |  |
| 100 | The specification shall support interactions where it will be rendered to a certain group of users. | Some support in SD needed. Other system spec TBD. |  | Phase 3 |  |
| 100.1 | "The specification shall support privacy protection features related to the delivery of client specific metadata (e.g. position, orientation, etc.), especially for interactions related to bi-directional delivery applications (e.g. social VR).  NOTE: This requirement needs to be further clarified. |  |  | Phase 3 |  |
| Use Case Specific Requirements | | | | | |
| Social VR | | |  |  |  |
| 101 | The specification shall support interactions in case of capability mismatching of user devices. | Some support in SD needed. Other spec TBD. |  | Phase 3 |  |
| 102 | The specification shall support that a user can recognize objects to interact with. |  |  | Completed with interactivity extension, and application specific implementation |  |
| 103 | The specification shall support metadata that provides the position/orientations of remote users in the immersive environment of a local user |  |  | Completed with the avatar extension at the node level which provides spatial transformation metadata |  |
| 104 | The specification shall support metadata that provides the direction of view of the local user in its immersive environment |  |  | Partially completed with avatar extension.  Clarify the requirement with pose metadata (completed in MAF) and gaze metadata (not completed – phase 3) |  |
| 105 | The specification shall support metadata that indicates which remote user is being looked at by the local user |  |  | Partially completed with avatar extension.  Clarify the requirement with pose metadata (completed) and gaze metadata (not completed – phase 3) |  |
| 106 | The specification shall support metadata that indicates whether a video stream corresponds to user A looking straight into the camera (main camera) or not (side camera) |  |  | Phase 3 |  |
| 107 | The specification shall support containers (e.g. ISOBMFF) to carry the above metadata to remote users. |  |  | Partially completed in system for V3C stream. |  |
| 108 | The specification shall support metadata that provides x/y/z and yaw/pitch/roll coordinates that provide the position/orientation of a camera or audio source with respect to the perceived immersive environment. | supported by gltf |  | Completed |  |
| 109 | The specification shall support metadata that provides the type and other details of the camera or audio source. |  |  | Completed with audio spatial semantic in Ed1 |  |
| 110 | The specification shall support metadata to identify and distinguish cameras and audio sources. | support with legacy glTF camera metadata. And audio spatial semantic SD extension |  | Completed with audio spatial semantic in Ed1 |  |
| 111 | Metadata to signal details of the visual and/or auditive/haptic indication of the camera/audio source position/orientation in the immersive environment |  |  | Completed with spatial audio extension in Ed1, haptic extension in AMD2 |  |
| 112 | Containers (e.g. ISOBMFF) to carry the above metadata to the other user(s). | MPEG System FF |  | Not SD related  Partially completed for V3C media. Phase 3 for others |  |
| 113 | Metadata that provides the angle of view of the camera | Seems gltf |  | Completed |  |
| 114 | Metadata that provides the distance between the camera and the video-captured person  Note: same requirement than 96.1 | Can be calculated by the Presentation Engine based on spatial attributes of the scene graph |  | Completed. |  |
| 115 | Metadata about the position of the video-captured person in the video frame |  |  | Completed with video texture extension |  |
| 116 | Containers (e.g. ISOBMFF) to carry the above metadata to remote users. | MPEG System FF |  | Not SD related |  |
| 117 | The specification shall support positioning multiple objects in the same omnidirectional environment where these objects are represented by an image or a video Note: the primary use case of this requirement is to place multiple users in, e.g., a video conferencing setting in an omnidirectional environment, together with other elements, e.g., a video screen. |  |  | Completed with Amd1 |  |
| 117.1 | The specification shall enable making the omnidirectional environment consistent: |  |  | Phase 3 |  |
| 117.1.1 | - for all users that are embedded in their omnidirectional environment |  |  | Phase 3 |  |
| 117.1.2 | -  between the users (they can look at one another, and can see when another person is looking at them) |  |  | Phase 3 for gaze support and additional metadata |  |
| 117.1.3 | -  for multiple users looking at / pointing at a common element in the omnidirectional environment (e.g., a video screen) |  |  | Phase 3 for gaze support and additional metadata |  |
| 117.2 | The specification shall support synchronization of user viewpoints and orientations (i.e. where each user is looking at), as well as content playback status, between users. Note: these individual environments need not necessarily be the same, as long as they are internally consistent for all participants individually Note: “consistent” means the right visual perspective for all objects as well as audio/visual alignment |  |  | Phase 3 for gaze support |  |
| 117.3 | The specification shall enable bringing multiple users together in the same omnidirectional environment even when they are captured using their own individual 3DoF coordinate system |  |  | Phase 3 |  |
| 117.4 | The specification shall enable positioning a rectangular object (e.g., a video screen) inside the omnidirectional environment with correct perspective and parallax. The surface of this object may be at an arbitrary distance from the user, and may have an arbitrary orientation in 3D space. |  |  | Phase 3 |  |
| Haptics Requirements - Haptics with 6 Degrees of Freedom | | |  |  |  |
| 118 | [Haptics Phase 1] The specification shall support synchronization of haptic effects with audio-visual content. |  |  | Completed in Haptic and system |  |
| 119 | [Haptics Phase 2] The specification shall support association of haptic feedback (tactile, kinesthetic, essence, texture) with a 2D, 3D, AR, VR, audio and/or video objects and environments. |  |  | Completed in SD AMD2 |  |
| 120 | [Haptics Phase 2] The specification shall support rendering of haptic feedback on multiple devices across multiple body locations |  |  | Completed in SD AMD2 and Haptic |  |
| 121 | [Haptics Phase 1 & Phase 2] The specification shall support high-fidelity, low-latency user tracking to enable active interaction and haptic feedback |  |  | Completed in SD AMD2 and Haptic |  |
| 122 | [Haptics Phase 2] The specification shall support dynamic generation and synthesis of haptic effects based on other metadata, media streams or external data sources. |  |  | Completed in Haptic |  |
| Requirements on Haptics Renderer | | |  |  |  |
| 123 | [Haptics Phase 1 & Phase 2] The specification shall support presentation of haptic media on alternate devices or devices which may have different performance characteristics from the mastering system. |  |  | Completed in Haptic |  |
| 124 | [Haptics Phase 1] The specification shall support mixing and modulation between and within haptic tracks. |  |  | Completed in Haptic |  |
| Haptics Delivery | | |  |  |  |
| 125 | [Haptics Phase 1] The specification shall support multi-channel haptic delivery. |  |  | Completed in Haptic |  |
| 126 | [Haptics Phase 1] The specification shall support compression of haptic data. |  |  | Completed in Haptic |  |
| 127 | [Haptics Phase 1 & Phase 2] The specification shall provide the necessary information to support transcoding. |  |  | Completed in Haptic |  |
| 128 | [Haptics Phase 2] The specification shall support specification of round-trip (input to feedback) latency requirement. |  |  | Not supported yet.  Haptic phase 2,  SD phase 3 |  |
| Haptic and interaction model | | |  |  |  |
| 129 | [Haptics Phase 2A] The specification shall support interactivity models related to avatar position and orientation. |  |  | Completed in SD AMD2 |  |
| 130 | [Haptics Phase 2B] The specification shall support coding and presentation of interactivity models related to avatar-scene or avatar-avatar interactions. |  |  | Partially addressed in SD AMD2. To be completed in SD phase 3 |  |
| 131 | [Haptics Phase 2] The specification shall support different media types and various haptic feedback paradigms (pre-rendered, synthesized). |  |  | Completed in Haptic |  |
| Spatial Computing Server requirements for eXtended Reality (XR) | | | | | |
| Requirements on SCS | | |  |  |  |
| 132 | The SCS shall create the XR Spatial Description of the XR space. | Seems internal processing? |  | Not SD related |  |
| 133 | The SCS shall update and fuse existing XR Spatial Description of the XR space. | Seems internal processing? |  | Not SD related |  |
| 134 | The SCS shall provide XR Spatial Description in a standard representational format (e.g. scene description), upon request of XR devices (UEs) on different platforms (desktop and mobile). |  |  | May need to Phase 3 |  |
| 135 | The SCS shall accept the XR spatial data (XR Features) captured by XR UEs over the network or direct connection. |  |  | Phase 3 |  |
| 136 | The SCS shall deliver the XR Spatial Description to XR UEs over the network or direct connection. |  |  | Phase 3 |  |
| 137 | The SCS shall convert other representation formats to XR Spatial Description. |  |  | Phase 3 |  |
| 138 | The SCS shall support global registration of indoor-only XR Spatial Description and provide means for managing indoor XR spatial configurations with real geo-coordinates. |  |  | SD support Completed in AMD2 (support of XR anchor with geospatial coordinates) |  |
| XR Spatial Description | | |  |  |  |
| 139 | The Description shall contain XR features for indoor and/or outdoor localization/tracking purpose. | The requirement may include retrieving information on sensor and users |  | Partially completed in AMD2  To be continued in Phase3. |  |
| 140 | The Description should have a data structure to represent the XR space with appropriate pixel-to-meter scaling factor to the physical world. |  |  | Completed in AMD2 (support of XR anchor) |  |
| 141 | The Description shall allow the definition of the XR space from small scale space like indoor environment to large scale outdoor space. |  |  | Completed in AMD2 (support of XR anchor) |  |
| 142 | The Description shall support visual and/or non-visual features (including audible and mechanical features) of the XR space for real-time localization and tracking (pose estimation). |  |  | Completed in AMD2 (support of XR anchor) |  |
| 143 | The Description shall allow navigation of the XR client from one space to another when multiple indoor XR spaces are used. | Gltf allows for multiple scene |  | partially completed.  Some potential work in Phase 3. |  |
| 144 | The Description shall include XR Anchor objects with presentation properties as the placement for XR Media object. |  |  | Completed in AMD2 (support of XR anchor) |  |
| 145 | The Description should support stationary and moving (mobile) XR anchors (e.g. for avatars). |  |  | Completed in AMD2 (support of XR anchor) |  |
| 146 | The Description should keep a mapping between virtual coordinate system to one or multiple physical coordinate systems for different space. |  |  | Completed in AMD2 (support of XR anchor) |  |
| XR Spatial Description | | |  |  |  |
| 147 | The position for media object in the XR space shall be defined by an XR anchor. |  |  | Completed in AMD2 (support of XR anchor) |  |
| 148 | There shall be an association defined for logic links between XR anchors and live streaming media object. |  |  | Completed in AMD2 (support of XR anchor) |  |
| 149 | There shall be an association defined for logic links between XR anchors and on demand media object. |  |  | Completed in AMD2 (support of XR anchor) |  |
| 150 | The Media shall support any type of media content (be media agnostic), with MPEG codecs and MPEG formats. |  |  | Phase 3 for MPEG immersive audio |  |
| Requirements on XR Client | | |  | Mostly out of scope of SD |  |
| 151 | The XR Client shall capture environment visual data (image, video, or other visual cues), and/or non-visual features such as environment audio data (audio sources, proximity, or other audio cues), environment location data (from IMU and GPS sensors), and environment object data (geometry, surface and material characteristics, proximity, or other object cues), that is, the XR features; and generate a collection of XR features (for the privacy concern) at multiple locations within the environment. |  |  | SD phase 3 support for gaze capture, and environmental data (sensor) (e.g. how to interact with sensor data) |  |
| 152 | The XR Client should capture its spatial (local or global) location(s) through the sensors, if available. |  |  | Mostly out of scope of SD |  |
| 153 | The XR Client should generate a new XR spatial description and upload it to the SCS for further processing (SCS-002) over the network or direct connection. |  |  | API and SD spatial description uplink  phase 3 |  |
| 154 | The XR Client should update an existing XR spatial description received from SCS and upload it to the SCS, if feasible |  |  | API and SD spatial description uplink  phase 3 |  |
| 155 | The XR Client should load new XR Descriptions when navigating from one physical place to another (with the help of 144). |  |  | To be addressed in phase 3 with interactivity/ event-based udpate |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |