IBC2023

MPEG-I SCENE DESCRIPTION: A DYNAMIC SCENE DESCRIPTION FRAMEWORK FOR IMMERSIVE MEDIA

T. Stockhammer[1], I. Bouazizi[1], M.-L. Champel[2], E. Potetsianakis[2], E. Thomas[2], L. Kondrad[3], E. Alexiou[4], G. Martin-Cocher[5], G. Bhullar[5], Q. Avril[5], Q. Galvane[5], J. Regateiro[5], P. Hirtzlin[5]

[1]Qualcomm, [2]Xiaomi Technology, [3]Nokia Technologies, [4]TNO, [5]InterDigital

ABSTRACT

Immersive media applications offer experiences which immerse the user in a virtual or hybrid environment and offer more degrees of freedom than with traditional 2D video content. Platforms providing immersive media often enable the user to interact with the content and/or with other users in shared virtual or mixed reality spaces.

To address the need for an interoperable cross-platform exchange format and interactive solution for such 3D environments, ISO/IEC JTC 1/SC29/WG03 MPEG Systems has standardized a Scene Description framework in ISO/IEC 23090-14 [1], that serves as an entry point format to compose rich 3D scenes, referencing and positioning 2D and 3D assets in the scene, blending with the real world, rich interactivity and providing real-time media delivery.

Carriage formats have also been defined for the delivery of the Scene Description data and of the linked assets, based on the well-known and ubiquitous ISOBMFF standard i.e., ISO/IEC 14496-12 [2].

INTRODUCTION

Immersive media applications offer experiences, where the user is immersed into virtual or hybrid environments. The user is able to experience the content in 3D and enjoy more degrees of freedom compared to traditional 2D content. Platforms providing immersive media also often give the user the ability to interact with the content and/or with other users, in shared virtual spaces.

Immersive media is becoming increasingly prevalent and start to influence the way we work and entertain ourselves. Immersion is achieved by introducing the depth dimension in media modalities (visual and auditory) traditionally digitally expressed in a 2D fashion. The trend of transition from 2D to 3D media was initially started by Virtual Reality (VR), mainly driven by the availability of affordable VR headsets. However, unique Augmented Reality (AR) and Mixed Reality (MR) immersive experiences are also becoming popular, supported by the release of consumer devices, such as see-through Head Mounted Displays (HMD) and glasses. A number of immersive experiences are also achievable on smartphones.

One of the key technologies in enabling immersive media user experiences is a scene description. Scene description defines the structure and composition of a 3D scene, referencing and positioning the 2D and 3D assets in the scene, and provides all necessary information that can be used by an application to render the 3D scene properly to an end-user.

The need for a solution to enable cross-platform exchange and interaction in 3D environments became evident and a number of forums and Standards Developing Organizations (SDOs) started to define the needed technology. ISO/IEC Moving Picture Experts Group (MPEG) Working Group 3 (WG03) defines a scene description framework in part 14 of the MPEG-I series of standards (i.e., ISO/IEC 23090-14), serving as an entry point to rich 3D dynamic and temporal scenes, enabling immersion, fusion with the real world and rich interactivity, while providing real-time media and scene update delivery.

Furthermore, the standard defines an architecture together with an application programming interface (API), that allows the application to separate access to the immersive timed media content from the rendering of this media. The separation and the definitions of this API allow the implementation of a wide range of optimization techniques, such as the adaptation of the retrieved media to the network conditions, partial retrieval, access at different levels of detail, and adjustment of the content quality.

This article provides an overview of MPEG-I Scene Description (MPEG-SD) and it is organized as follows.

The first section describes the architecture framework utilized in MPEG-SD, which is followed by a section describing all the new features introduced by the first edition of the standard and its amendments. After that we provide information of the storage and transport aspects related to the MPEG-SD. The last three sections look at future standardization projects related to MPEG-SD, future work, and conclusions.

FRAMEWORK ARCHITECTURE

Due to its complexity, immersive media cannot be effectively consumed using traditional 2D client architectures, and it comes with a set of new requirements. In MPEG-SD, a reference architecture framework was defined, and is presented in Figure 1.

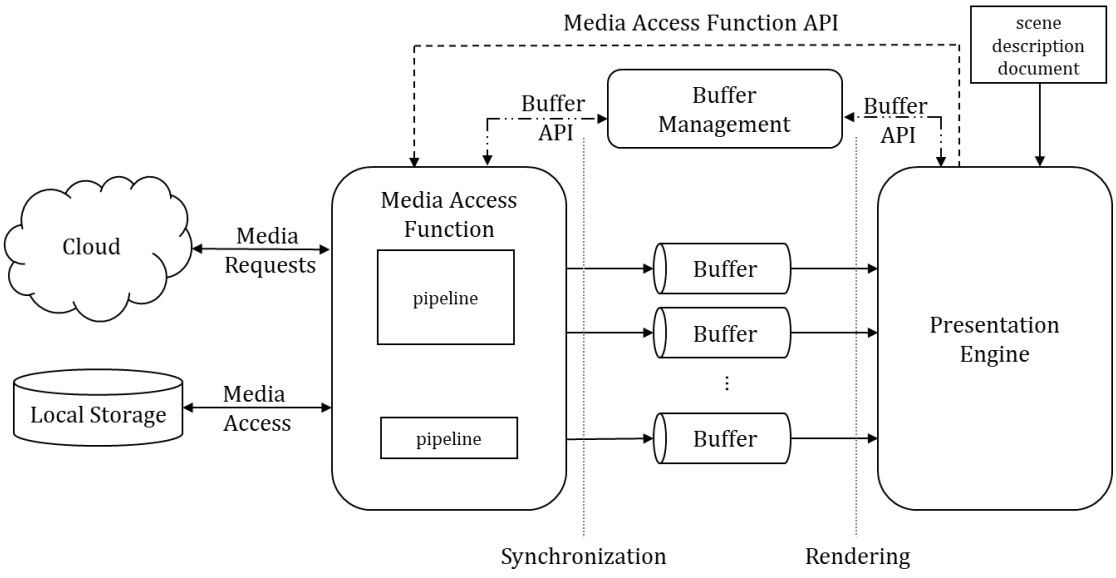

Figure 1: MPEG-SD reference architecture.

In the proposed reference architecture for immersive media, a traditional 2D media player is replaced by a Presentation Engine. The Presentation Engine is the core of immersive applications and it is responsible for multi-modal rendering of a scene which may be composed of audio, visual, and haptics media. To keep the focus of the Presentation Engine on high fidelity rendering, the architecture includes another component, the Media Access Function (MAF), which is responsible for the media access and processing functions. The Presentation Engine delegates the handling of the media to the MAF, which constructs a suitable media pipeline to transform the media from a delivery format into formats that are used during the rendering process in the Presentation Engine. The MAF uses information such as the MIME type and codec parameters to identify support for the media reconstruction and assemble the proper media pipeline. For example, a media pipeline created by the MAF could perform fetching, decoding, decryption, and post-processing of the referenced media. The Presentation Engine uses the MAF API to request the immersive media in the scene. Along with the media sources, the Presentation Engine provides information about the current viewer's and the object's pose to allow the media pipeline to optimize the delivery. For instance, the media pipeline may adjust the level of detail for the selected media based on the object distance and visibility to the viewer.

In the pipelines created by the MAF, the processed media is fed into buffers with a clearly defined format that is signalled through scene description document defined in ISO/IEC 23090-14. The buffer formats signalled in the scene description document provide a clear interface between the MAF and the Presentation Engine.

FEATURES

One of the well established and widely used scene description formats is the Khronos glTF 2.0 specification, released in 2017 [3]. glTF 2.0 is a file format specification designed for the efficient transmission of 3D scene and for efficiently loading and representing these scenes and 3D models in applications at runtime. The glTF 2.0 scene structure is described with JSON, a compact human readable Javascript Object notation, that is easily parsable. The glTF 2.0 describes a) the scene structure, in essence a hierarchy of nodes that defines a scene graph and b) the 3D objects stored and used in the scene, which are referenced by the scene nodes and defined by meshes. glTF defines some "properties", notably Materials (appearance of objects), Animations (transformation over time of 3D objects), Cameras (view configurations) and more. This format is extensible and provides a solid and efficient baseline for exchangeable and interoperable scene descriptions toward resource constrained devices such as see-through HMD and glasses. However, the main specification of glTF still contains gaps in relation to the timed media. To address those gaps that are required for immersive applications, MPEG-SD defines new features using glTF extension mechanisms. The core set of the new features is provided in the first edition of the ISO/IEC 23090-14, while additional functionalities is under development that will be released in the two amendments of ISO/IEC 23090-14.

The Amendment 1 of ISO/IEC 23090-14 provides support for the Visual Volumetric Video-based Coding (V3C) standard that was developed by MPEG. V3C is a generic mechanism used by applications targeting volumetric video content, such as point clouds, immersive video with depth, mesh representations of visual volumetric frames, etc. Currently, ISO/IEC 23090-5 [4] specifies the applications of V3C targeting Video-based Point Cloud Compression (V-PCC) representations of visual volumetric frames in Annex H of ISO/IEC 23090-5, and ISO/IEC 23090-12 [5] specifies the applications of V3C targeting MPEG Immersive Video (MIV).

The Amendment 2 of ISO/IEC 23090-14 provides support for immersive audio, anchoring, interactivity, lighting, avatar, and haptics.

The features mentioned above are specified as a collection of extensions to glTF 2.0 [6]. All MPEG-SD glTF extensions specified in the first edition of the ISO/IEC 23090-14 have been registered in Khronos as vendor extensions with the support ofthe Khronos glTF working group. The registration of MPEG-SD glTF extensions defined in Amendments 1 and 2 is ongoing.

First Edition

ISO/IEC 23090-14 first edition defines, along with the architecture framework, a first set of vendor extensions to Khronos glTF 2.0 that addresses the requirements identified as essential for the distribution of real-time immersive media content. The extensions can be separated into two groups.

The first group is formed with extensions that enable the timed framework. It includes: MPEG_media, MPEG_accessor_timed, MPEG_buffer_circular.

The second group of extensions utilises the timed framework, and builds on top of the first group enabling the inclusion of dynamic, temporal media. It includes: MPEG_texture_video, MPEG_audio_spatial, MPEG_scene_dynamic, MPEG_viewport_recommended, MPEG_mesh_linking, MPEG_animation_timing.

The set of extensions and their placement in the node hierarchy is depicted in Figure 2.

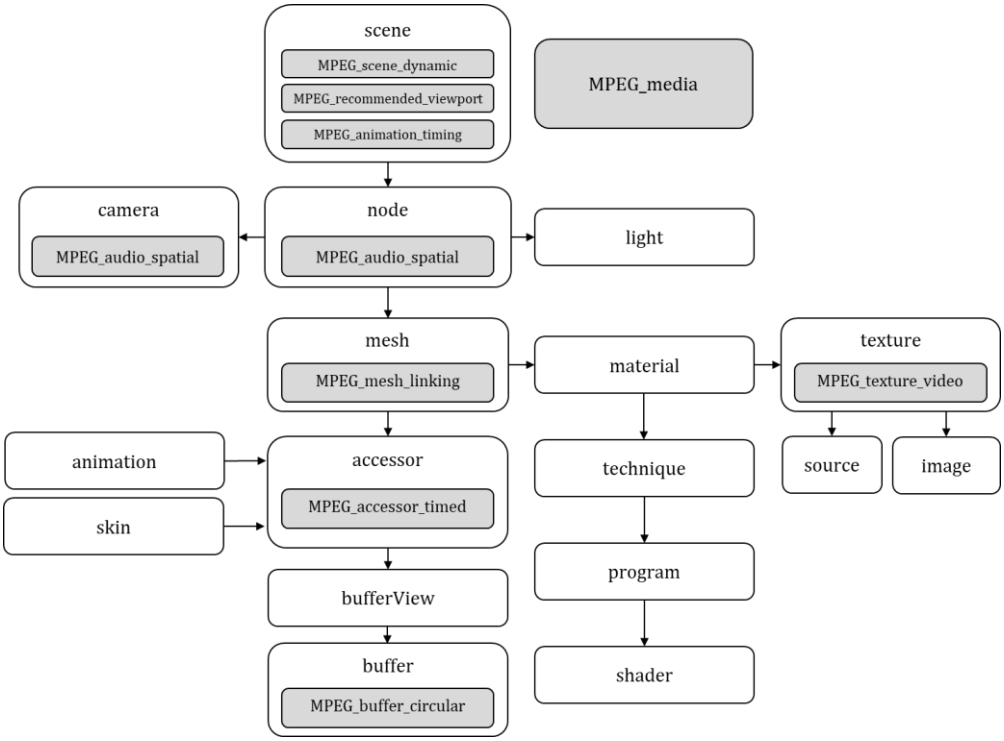

Figure 2: First edition extensions framework description.

Each extension is described below.

Media (MPEG_media)
An extension listing external media that are referenced in a scene description document by other elements. The extension allows creators to describe the content in alternative formats,

thus offering the ability to select the most appropriate format to access based on the application's capabilities.

Timed accessor (MPEG_accessor_timed)
An extension indicating to an application that the media described by the accessor element is timed. Additionally, the extension may signal to the application the presence of dynamic fields that change on a frame-to-frame basis.

Circular buffer (MPEG_ buffer_circular)
An extension indicating that a circular buffer is used for passing timed media, with concurrent read/write access to the buffer. The extension also links the underlying buffer to a media described by MPEG_media extensions. The buffer extension is used to pace the access to the buffer, compensating for any differences between the media flow through the media pipeline in the MAF and the rendering process in the Presentation Engine. The extension also indicates that this extended buffer does not have the original glTF 2.0 restriction of containing only static data.

Video texture (MPEG_texture_video)
An extension that provides a mechanism to provide a dynamic texture by channelling the texture data through buffers described by MPEG_accessor_timed and MPEG_buffer_circular extensions.

Spatial audio (MPEG_audio_spatial)
An extension that offers support for spatial audio in a 3D scene by introducing audio sources, reverbs, and an audio listener. The first denotes input audio in the scene, the second denotes effects applied to the input audio, and the last is the output audio. The audio sources reference timed accessors.

Animation timing (MPEG_animation_timing)
An extension that enables alignment between media timelines and animation timelines defined by glTF 2.0. Using this extension narrated stories could possibly be created. The animation timing metadata could allow simultaneous pausing and other manipulation of animations defined in glTF 2.0 and media.

Recommended viewport (MPEG_recommended_viewport)
An extension that provides a stream of sparse samples that indicate the author recommended pose for viewing the scene. This might especially be useful in cases where the scene is consumed on a 2D display device.

Mesh linking (MPEG_mesh_linking)
An extension to link two meshes and provide the mapping information. This extension allows, for instance, to apply animations to dynamic meshes with changing topologies by defining it as a dependent mesh on a shadow mesh with a static topology. The transformations and required metadata are defined for the shadow mesh and are transferred to the dynamic mesh by using the mapping information defined within the extension.

Scene update (MPEG_scene_dynamic)
An extension that allows to indicate that the scene description document may be updated. Scene updates are expressed as a scene description document or as a patch document using the JSON Patch protocol [7].

Amendment 1

Supporting content coded with Visual Volumetric Video-based Coding (V3C)

The first amendment of the MPEG-SD introduced the integration of V-PCC and MIV codecs which are based on a common V3C scheme and are part of the MPEG-I collection of standards. Two variety of pipeline structures were defined. A codec-independent pipeline (i.e. pipeline in Figure 3a) is such where the final reconstructed output i.e. point cloud is generated by the MAF. The final reconstructed output is shared with the presentation engine for rendering. On the other hand, codec-dependent pipeline (i.e. pipeline in Figure 3b) is designed to leverage GPU resources at the presentation engine. An intermediary format for content coded with V3C scheme is defined. The MAF is responsible to decode a V3C content and generate the intermediary formats for different components of the V3C stream. The Presentation Engine is responsible to generate the final reconstruction of the 3D content with the intermediary V3C format as an input.

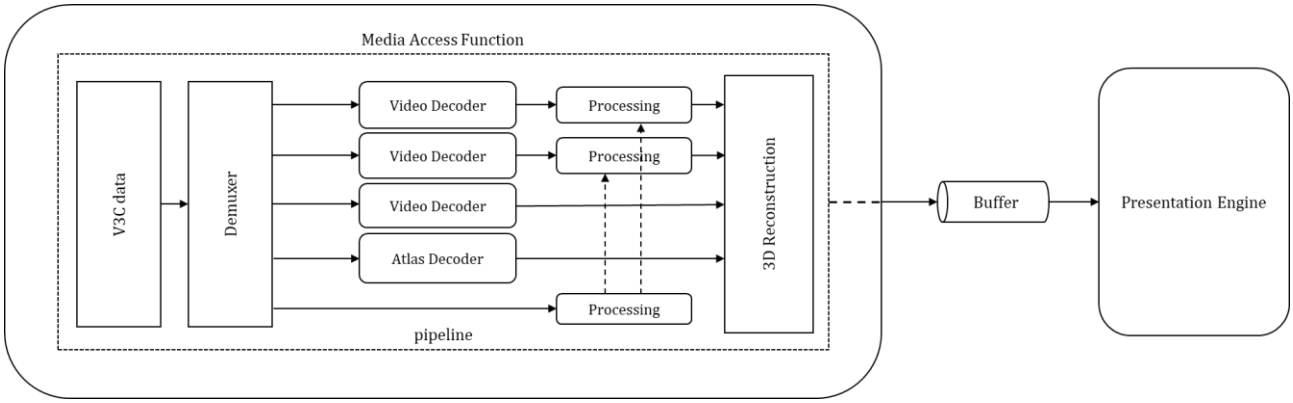

Figure 3a: Codec-independent pipeline mechanism to process a V3C content.

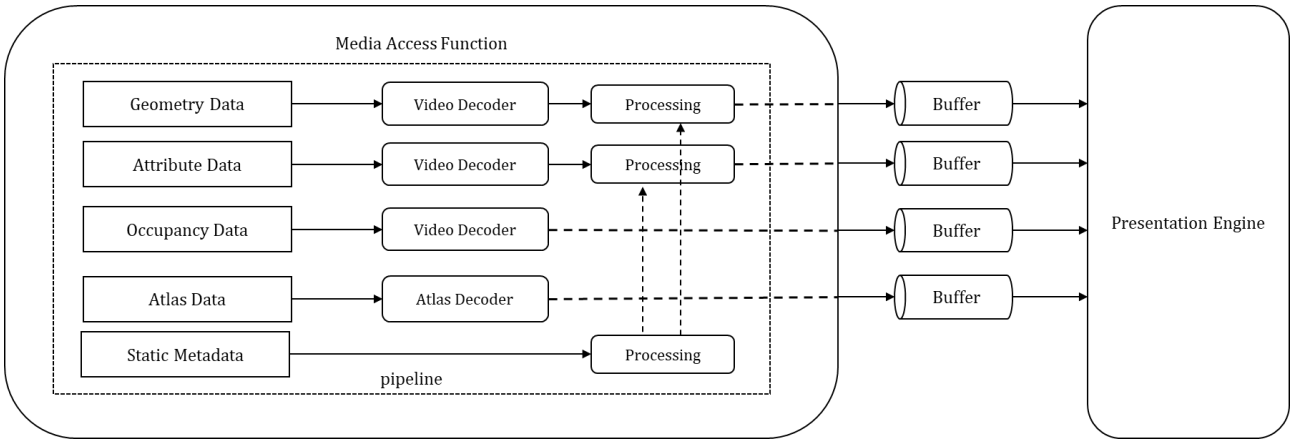

Figure 3b: Codec-dependent pipeline mechanism to process a V3C content**.**

Sampling a YCbCbr texture

A mechanism to signal a YUV texture is introduced with a YCbCr sampler extension. The YCbCr sampler extension follows the modern graphics specification such as Vulkan specification to describe a YCbCr texture. The extension provides relevant configuration information to read and sample a YCbCr texture.

Amendment 2

The second amendment of MPEG-SD defines a set of vendor-extensions to Khronos glTF 2.0 that address the requirements for the distribution of real-time immersive and interactive media content.

For that purpose, a collection of new extensions are introduce, including: MPEG_scene_anchor, MPEG_scene_interactivity, MPEG_node_anchor, MPEG_node_interactivity, MPEG_node_avatar, MPEG_light, MPEG_haptic and MPEG_material_haptic.

The set of the extensions and their placement in the node hierarchy is depicted by the following Figure 4 and individually described below.

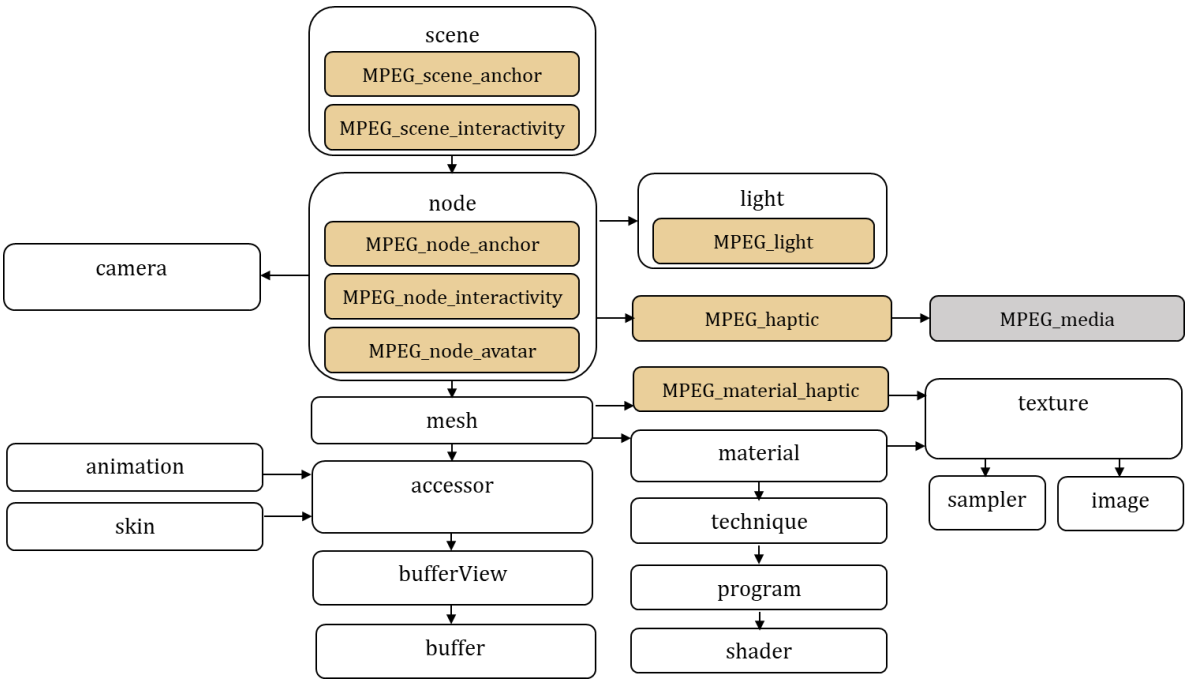

Figure 4: Amendment 2 extensions framework description.

Augmented Reality anchor (MPEG_scene_anchor, MPEG_node_anchor)
To support AR experiences where virtual content is seamlessly inserted into the user's real environment, MPEG-SD provides to the content creator the possibility to describe the spatial relationships between the virtual objects and particular real locations. As shown in Figure 5, the pose of a virtual asset is defined in the local space of an AR anchor. The AR anchor corresponds to a fixed position in the XR space of a trackable, i.e., an element of the real environment of which feature are available and/or could be extracted. To support a variety of indoor and outdoor AR experiences, several types of trackable are defined such as, controller-based, horizontal, or vertical planes, 2D or 3D marker, geospatial coordinates.

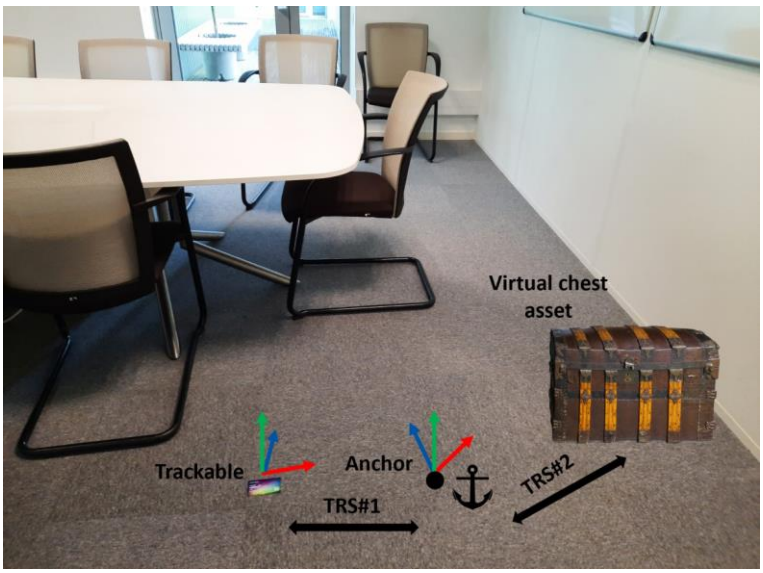

Figure 5: Spatial relationships between a trackable, AR anchor and virtual asset.

Interactivity (MPEG_scene_interactivity, MPEG_node_interactivity)
A flexible and generic framework is provided for describing interactivity at runtime. Both the interactions between the user and the virtual objects and between the virtual objects themselves are supported. This framework is based on the definition of behaviors, composed of a logical combination of triggers and the related actions to launch sequentially or in parallel when the triggers activation status is met. Generic triggers are defined based on proximity, visibility, collision or user input condition. A set of actions is also defined to modify the pose and the properties of virtual objects, to control MPEG media sources (i.e., audio, video, haptics) and animation. The framework also allows to provide parameters to control the physics simulation with collision handling.

Haptics (MPEG_haptic, MPEG_material_haptic)
The Amendment 2 on MPEG-SD provides haptics support. This support of haptics is based on the MPEG standard for the Coded representation of Haptics - Phase 1. Haptic information can be included in the scene at two different levels: it can be attached to a node or it can be attached to a mesh. The first extension allows to attach haptic data directly to a node by referencing MPEG_haptic medias. The second extension allows a more accurate description of the haptic properties of an object by attaching the haptic data directly to a mesh using haptic textures. This haptic information can be used directly to render timed haptic data or it can be used with the interactivity framework to provide interactive haptic feedback.

Avatar (MPEG_node_avatar)
The current ongoing Amendment 2 specification of MPEG-SD provides support to an avatar MPEG reference model and a glTF node extension. The reference model can be edited to create new avatars, or other avatars can be added just by using the node extension. The current framework for avatars has the objective of permitting a generic representation of 3D avatars so it facilitates interactivity with objects in a 3D shared environment.

Even though in MPEG-SD avatars are defined in a generic way to cover a wide range of use cases, they can also be interpreted as a 3D representation of users in 3D shared

environments i.e., human avatar representation. The definition of avatar within MPEG has been initiated in MPEG-V [8] and currently adopted in MPEG-SD through the more recent glTF format. An example integration of avatars is specified in MPEG-SD by the MPEG reference model in the Annex H of the ongoing Amendment 2.

In the Annex H, the visual elements and the animation set-up of the MPEG-SD humanoid reference avatar are described in more detail. The MPEG-SD humanoid reference avatar is named *Morgan* (it stands for **M**PEG **O**riginal **R**eference **G**eometric **A**vatar **N**eutral) and is composed of:

- a complete body base mesh topology (with three Levels of Details (LoD)),
- body part semantic labels,
- a skeleton (with a corresponding hierarchy) and a set of skinning weights,
- a set of facial blend shapes and facial landmarks,
- a set of additional geometries, such as eye globes, jaws, teeth and tongue.

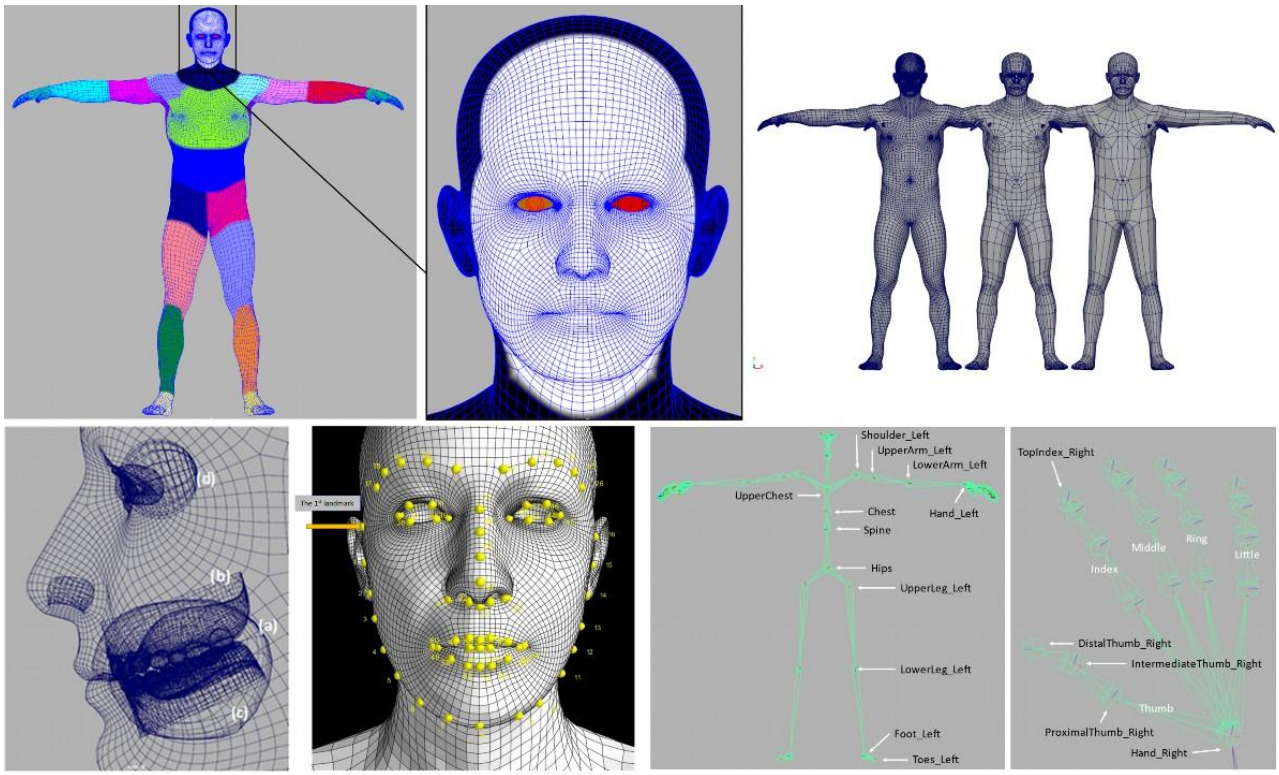

Figure 6: From top left to bottom right: (1) Front view of body parts semantic areas - (2) Morgan's face - (3) 3 different LOD - (4) internal facial details - (5) 68 facial landmarks - (6) Morgan's skeleton.

The Annex H provides a detailed table describing the correspondence between vertices' and faces' ranges and the all associated semantic labels. The LoD of the Morgan avatar is described as low, medium and high resolutions, and each is accompanied by different mesh and skeleton topologies. As part of the Morgan avatar, a basic glTF format description is available to visualise the basic geometry and to encode the hierarchical skeletal structure.

The glTF format specification in the ongoing Amendment 2 specification of MPEG-I SD defines a node extension MPEG_node_avatar, as illustrated in the Figure 6. The MPEG_node_avatar extension at the current state is a signalling value that informs whether the node is an avatar or not ("True" or "False").

Finally, ongoing work in MPEG on interactivity is also aiming to define interactivity (e.g. actions) between avatars and/or between an avatar and the scene.

Lighting (MPEG_light)
When inserting virtual information in a captured real-world environment, lighting is a fundamental cue for providing a realistic experience to the user. Improper lighting and shadows can break the immersive illusion of the inserted virtual objects. In a VR context, accurate lighting models allow to achieve a high level of realism. In the MPEG-SD lighting extensions, the light sources can be of two natures: real or virtual. In both cases, the light sources may be represented by the same mathematical models (punctual light, ambient light, etc.). The common model to reconstruct realistic lighting, which is reused in MPEG-SD, is composed of three elements:

1. Main directional light which represents the main light source. It can be used to cast shadows.
2. Ambient spherical harmonics which represents the remaining ambient light energy in the scene.
3. An environment cubemap (texture) which can be used to render reflections in shiny metallic objects.

As of now, there exist several lighting extensions for glTF 2.0 as part of the official Khronos glTF repository that are EXT_lights_image_based, KHR_lights_punctual and EXT_lights_ies. Compared to those extensions, MPEG-SD lighting extensions mainly adds two functionalities on top of those extensions. The first one is the ability for a scene creator to describe lighting information varying over time in the scene. For image-based lighting, the scene creator can use a video encoded sequence where each frame is an environment cubemap. This way, the ambient lighting information evolves over time based on the timing information in the video file. For punctual lights, the MPEG lighting extension provides accessors where timed samples containing the light source information can be accessed. The second main functionality is the ability to have localised lighting information in the scene instead of solely global information. This allows richer and more complex scenes, for instance describing several rooms in a building.

STORAGE AND TRANSPORT FORMAT

ISO/IEC 23090-14 defines carriage formats related to delivery of scene description data as well as to delivery of data related to glTF 2.0 extensions. The carriage format is based on the ISO/IEC 14496-12 ISOBMFF standard.

To facilitate delivery of the scene description to a client, ISO/IEC 23090-14 defines how glTF description files and related data can be provided as non-timed and timed (i.e. as samples of track) data encapsulated in an ISOBMFF file.

A number of extensions, MPEG_scene_dynamic, MPEG_mesh_linking, MPEG_animation_timing, indicate that a particular form of timed data is provided to a Presentation Engine during the consumption of the scene and the Presentation Engine shall act based on the changing information.

The MPEG_media extension enables the referencing of external media streams that are delivered over protocols such as RTP/SRTP, MPEG-DASH, or others. In order to allow addressing media streams without actually knowing the values for the protocol scheme,

hostname, or port, ISO/IEC 23090-14 defines a new URL scheme. The scheme requires the presence of a stream-identifier in the query part, however, it does not dictate a specific type of identifier. It allows for the usage of the Media Stream Identification scheme (RFC5888), a labeling scheme (RFC4575), or a 0-based indexing scheme.

FUTURE WORK

MPEG has a number of technologies under consideration for future releases of MPEG-SD. Among them:

Advanced interactivity: this extension addresses the need to support scene updates generated at runtime from the activation of triggers. The advanced interactivity extension is typically required in the case of a common virtual scene, shared between multiple users, each of them interacting and modifying the scene at runtime.

Advanced scene understanding: this extension aims at providing advanced real-world representation in scene description to achieve a seamless integration of the virtual into the real world (an example may be the interaction of virtual objects falling on a real surfaces).

MPEG-I immersive audio: this extension aims at providing advanced audio immersion by defining acoustic properties for the virtual scene and objects.

CONCLUSIONS

In this paper, we described the motivations behind the work of the MPEG Systems Working Group on Scene Description and its principles, including the architecture framework, key extensions and formats. Currently, the MPEG Systems Working Group is progressing towards the completion of the Amendment 2 and is evaluating possible future use cases and needs for advance immersive scene-based media applications.

REFERENCES

[1] ISO/IEC 23090-14 Information technology — Coded representation of immersive media — Part 14: Scene description https://www.iso.org/standard/80900.html

[2] ISO/IEC 14496-12:2022 Information technology — Coding of audio-visual objects — Part 12: ISO base media file format

[3] glTF 2.0 specification https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html
[4] ISO/IEC 23090-5, Information technology — Coded Representation of Immersive Media — Part 5: Visual Volumetric Video-based Coding (V3C) and Video-based Point Cloud Compression (V-PCC)
[5] ISO/IEC 23090-12, Information technology — Coded Representation of Immersive Media — Part 12: MPEG immersive video
[6] glTF 2.0 Extension Registry
https://github.com/KhronosGroup/glTF/tree/main/extensions
[7] IETF RFC 6902
[8] ISO/IEC 23005 MPEG-V, Information technology — Media context and control