MPEG-I Scene Description
Overview

Imed Bouazizi
MPEG-I Target Use Cases

- Immersive Gaming
- Immersive Entertainment
- Immersive Collaboration
- Immersive Conferencing
Last Mile vs Interchange

**Last Mile**
- Low Complexity
- Flat Hierarchy
- Compressed components
- Adaptive and network friendly
- Support for Texture/Light Baking

**Interchange**
- High fidelity
- Superstructure
  - Hierarchical
  - Distributed
  - Preserves author’s intents/choices
  - Documents authoring process
- Lossless
- Preserves asset’s metadata/versioning

[Logos for GLTF, USD, ALEMBIC, and Maya]
ISO/IEC 23090-14 - MPEG-I Scene Description

• MPEG decided to extend an existing format rather than starting from scratch
  • No BIFS, MPEG-7, LASeR, etc. anymore

• glTF was recognized by MPEG as the best candidate for a baseline
  • Suitable for distribution/last-mile
  • Wide support with importers/exporters and renderers available in all environments

• The following gaps were identified:
  • No support for Audio
  • No support for timed media (dynamic meshes/point clouds, video textures, ...)
  • No support for scene updates
  • No decoupling of media access from rendering

• MPEG defined extensions to address these gaps
  • All extensions use the MPEG namespace

• Media Access Function (MAF) API
  • Decoupling Rendering from Fetching
  • Media Access Function offers API to the Presentation Engine to request media
  • Media and Metadata is fed into the Presentation Engine through buffers
Scene Description in MPEG-I Architecture

- Presentation Engine
  - Local Camera Input
  - Local Microphone Input
  - 2D/3D Media
  - Scene Graph
  - Scene Graph Update
  - User Input
  - Sensor Input

Flow of information:
- \( i-l \) for local input
- \( i-m \) for media input
- \( i-s \) for scene input
- \( i-i \) for integrated input
- \( t \) for time
MPEG-I Scene Description Architecture
MPEG-I Scene - Node Hierarchy

- **scene**
  - MPEG_scene_dynamic
  - MPEG_recommended_viewport
  - MPEG_animation_timing

- **camera**
  - MPEG_audio_spatial

- **node**
  - MPEG_audio_spatial

- **mesh**
  - MPEG_mesh_linking

- **accessor**
  - MPEG_accessor_timed

- **bufferView**
  - MPEG_buffer_circular

- **light**

- **material**
  - MPEG_texture_video

- **technique**

- **program**

- **shader**

- **texture**
  - source
  - image

- **animation**

- **skin**
MPEG_media Extension

- Top-level extension to glTF 2.0
- Allows referencing all types of media
  - Timed and non-timed
  - Compressed and non-compressed
  - MPEG and non-MPEG
- It supports different types of delivery
  - DASH & CMAF
  - WebRTC
  - HLS & CMAF
  - Local Storage (ISO BMFF, MP4)
- Orthogonal Functions: encryption, etc.
- This extension decouples Media Access Function from Presentation Engine in the Scene Graph
• **glTF accesses data through accessors**
  • They define the components of the data and their data types (e.g. a VEC3 of floats)
  • Semantics are provided by the referencing attribute/property (e.g. position)
  • The accessor points into a bufferView, which defines how the data is packed in the referenced buffer
  • No support for timed data

• **MPEG_accessor_timed**
  • Extension to accessor
  • Used to access all types of dynamic and timed media (audio, visual, volumetric, ...)
  • Backwards compatible: in case of no support, fallback to static data

• **MPEG_buffer_circular**
  • Extension to buffer
  • Dynamic variable-size swap chain buffer for exchange of media data for rendering
  • Acts as the interface between the Presentation Engine and MAF. All requested data through MAF API is delivered through a Buffer or Circular Buffer
  • Header is used to propagate metadata such as timestamps
  • Circular Buffer references MPEG_media
MPEG_texture_video

- Materials in the scene may make use of textures
- A texture in glTF 2.0 only supports references to images of format JPEG or PNG
- MPEG_texture_video adds support for dynamic textures such as atlases
  - Point into a timed accessor instead of an image
  - Keep the image pointer as fallback
  - To support dynamic atlases, texture coordinates themselves are dynamic and fed through a timed accessor as well
MPEG_audioSpatial

- glTF has no support for audio
- The MPEG_audioSpatial extension:
  - Audio Sources can be coupled to visual nodes to share the same transformations
  - Supports 3 types of nodes:
    - Audio Source: emits audio signals. Simple mono source and HOA source are supported
    - Audio Effect: a reverb zone effect is currently supported
    - Audio Listener: provides the position of the listener
  - The Audio Listener may be linked to the scene camera to allow for an immersive spatial experience. The listener will move together with the camera.
  - Actual rendering is not defined. It is up to the Audio Rendering Engine to convert the signals that are received at the audio listener into a format that matches the actual speaker setup. For example, binauralization is done for users wearing an HMD with headphones.
Media Access Function (MAF) API

- Support for wide range of formats through Plugins
- Endpoint for the Media Access Function API
- Optimized Media Fetching
  - Random spatial and temporal access
  - Partial delivery matching Presentation Engine needs
  - Integration with Cloud and Edge media processing
Media Pipelines

- The MAF instantiates and manages Media Pipelines
  - A media pipeline typically handles content of an attribute/component of an object/mesh
  - It produces content in the format indicated by the glTF file
  - The formatted frame is then pushed into the circular buffer

- Media Pipelines are highly optimized and customized for the type and format of media that is being fetched

- Media Pipeline maintains sync information (time and space) and passes that information as buffer metadata
Pipeline for Dynamic Meshes

Dynamic Mesh Pipeline

OBJ File Sequence Reader

Vertices
Texture Coordinates
Normals/Tangents/
Indices Data

Material Reader
HEVC Decoder for Texture

Processing
Buffer
Buffer
Buffer

Presentation Engine
Phase 2 Extensions - Interactivity

• Camera
  • Control the camera path
  • Viewer can only move within a constrained volumetric space along the camera paths

• Collision
  • Material characteristics
    • Friction
    • bounciness
  • Object may be static (immovable) or interactive (affected by collision)
  • Collision may trigger an animation
Phase 2 Extensions - Support for XR

- Allows users to anchor the virtual scene to their real environment
- Relative to a reference XR Space
  - Transforms, rotates, and scales the scene space onto the reference XR Space
- The reference XR Space maybe
  - A standardized XR space, e.g. stage, view, local
  - An application-defined space, e.g. bound to an object
- Uplink streaming of XR Actions