**3GPP TSG-SA WG4 Meeting #131 S4-250067**

**Geneva, CH, 17 – 21 February 2025 Revision of (S4aV250011)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.956** | **CR** | **pseudo** | **rev** | **-** | **Current version:** | **0.2.1** |  |
|  | | | | | | | | |
| *For* ***[HE](http://www.3gpp.org/3G_Specs/CRs.htm" \l "_blank)******[LP](http://www.3gpp.org/3G_Specs/CRs.htm" \l "_blank)*** *on using this form: comprehensive instructions can be found at  <http://www.3gpp.org/Change-Requests>.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | [FS\_Beyond2D] Representation Formats - Dynamic Mesh | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | China Mobile Com. Corporation | | | | | | | | | |
| ***Source to TSG:*** | SA4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_Beyond2D | | | | |  | ***Date:*** | | | 2025-02-07 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **B** |  | | | | | ***Release:*** | | | Rel-19 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The study item description in SP-240479 addresses the following objectives   1. Identify and document beyond 2D formats, that are market-relevant within the next years, generated from established and emerging capturing systems (including cameras for spatial video capturing), contribution, and usable on display technologies (smartphones, VR HMDs, AR glasses, autostereoscopic and multiscopic displays).   Dynamic meshes are are one of the most important and widely used beyond 2D representations of 3D assets. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | The revised document on dynamic mesh, updated to address comments from the VIDEO SWG Telco meeting:   * Ralf: What is the scenario? Single person? The size of content? … This has been requested for other formats.   + Jiayi: We propose it for the On-demand streaming scenario.   + Ralf: Do you intend also to bring additional test sequences for Meshes.   + Jiayi: Yes, but we are concerned about the licensing and the study timeline. I am trying to speed up the process. * Serhan: Several comments. About the references, the DM-3 is not the best source. There is another paper from Apple which is better. 2nd comment on 4.3.5.3, the APIs cannot be put in the same bucket. In 4.3.5.4, it would be good to have references in bullet 2, and in bullet 3, it would be good to complement this information. In 4.3.2.7.1 and 4.3.2.7.2, it is unclear.   + Jiayi: For the limitations in 4.3.2.7.2, I can change it. In 4.3.2.7.1, maybe we could remove the sentence. For other comments, I can check. * Thomas: Maybe PRY format has to be added.   + Jiayi: OK. * Thomas: In the description, do we exclude Avatar (base and animation) as a dynamic mesh?   + Jiayi: I don’t know if this is considered as a dynamic mesh or mesh with animation.   + Rufael: The connectivity can also differ.   + Ralf: If I think this is a MPEG definition (connectivity change), this should be documented.   + Thomas: That would be great to indicate it here and indicate connectivity as a superset.   + Jiayi: Maybe we can add the MPEG definition and what is used by the industry as another definition. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | One of the important and widely used beyond 2D representation format is not documented. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 4.3.5 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

# ===== CHANGE =====

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[DM-1] MPEG 136, CfP for Dynamic Mesh Coding, <https://www.mpeg.org/wp-content/uploads/mpeg_meetings/136_OnLine/w20972.zip>

[DM-2] MPEG 134, Use cases for Mesh Coding, <https://www.mpeg.org/wp-content/uploads/mpeg_meetings/134_OnLine/w20364.zip>

[DM-3] Y. Choi, J. -B. Jeong, S. Lee and E. -S. Ryu, "Overview of the Video-based Dynamic Mesh Coding (V-DMC) Standard Work," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 578-581, doi: 10.1109/ICTC55196.2022.9952734.

[DM-4] Information technology - Coding of audio-visual objects - Part 16: Animation Framework eXtension (AFX), ISO/IEC 14496-16.

[DM-5] Mammou, K., Kim, J., Tourapis, A. M., Podborski, D., & Flynn, D. (2022, September). Video and subdivision based mesh coding. In 2022 10th European Workshop on Visual Information Processing (EUVIP) (pp. 1-6). IEEE.

[DM-6] HS, Yang. and X. de Foy, "RTP Payload for V-DMC", Work in Progress, Internet-Draft, draft-hsyang-avtcore-rtp-vdmc-00, 18 October 2024, <https://www.ietf.org/id/draft-hsyang-avtcore-rtp-vdmc-00.html>.

[DM-7] ISO/IEC 12113:2022, Information technology — Runtime 3D asset delivery format — Khronos glTF™ 2.0, International Organization for Standardization, 2022.

[DM-8] Dynamic Mesh Documentation - Unigine Developer. Available at: https://developer.unigine.com/en/docs/latest/objects/objects/mesh\_dynamic/ (Accessed: 19 February 2025).

# ===== CHANGE =====

### 4.3.5 Dynamic Mesh

#### 4.3.5.1 Definition

A mesh is a structure composed of several polygons that define the boundary surface of a volumetric object. It typically includes five components: connectivity information, geometry information, mapping information, vertex attributes, and attribute maps. From MPEG, a dynamic mesh is defined as a mesh where at least one of these five components in varying in time [DM-1]. Such change can result from prescribed motion, flow induced rigid body motion, fluid structure interaction or adaptive mesh refinement. In the industry, a dynamic mesh is an object that represents a collection of vertices, edges and triangular faces (organized in polygons) defining the object's geometry that can be modified procedurally.(e.g., in an avatar’s facial expression or body movement) [DM-8].

Dynamic meshes are one of the immersive contents that are widely used in the commercial markets. For example, they can be used to represent 3D objects or digital avatar in VR/AR, digital twin city and etc. The demand for processing and visualizing such rich 3D content has led to the increasing popularity of dynamic meshes, as they are natively supported by virtually all the 3D software and graphic hardware, friendly to GPU rendering, and have a strong applicability to interactive and real-time 3D task [DM-3].

Many different formats can be used for storing dynamic mesh representation data. For example, the PoLYgon (PLY) format is introduced in section 4.6.3.5.2 of TR 26.928 [26.928], the OBJ file format is used in section 4.3.5.4.1.2, and also the glTF format as specified by the Khronos Group [DM-7].

MPEG has defined several use cases for dynamic mesh compression, including *Real-time 3D Immersive Telepresence*, *Content AR/VR viewing with Interactive Parallax* and *3D Free viewpoint Sport Replays Broadcasting* [DM-2]. The typical characteristics of the meshes in these use cases are summarized in Table. 4.3.5.1-1.

**Table 4.3.5.1-1 Typical Characteristics of Meshes in MPEG-Defined Use Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case** | **Triangle Count** | **Color Representation** | **Additional Properties** |
| **Real-time 3D Immersive Telepresence** | To represent a reconstructed human:  - 40,000 – 100,000 triangles (with color per vertex)  - 10,000 – 50,000 triangles (with texture maps) | - Texture maps: 2K–8K square pixels  - Color per vertex | - Normals and/or material properties for shader rendering |
| **Content AR/VR Viewing with Interactive Parallax** | To represent a reconstructed human:  - 10,000 – 100,000 triangles | - Texture maps: 2K – 8K square pixels  - Color per vertex | - Normals and/or material properties for shader rendering  - Global parameters for spatial constraints  - The meshes may be a part only of the total content transmitted |
| **3D Free Viewpoint Sport Replays Broadcasting** | To represent a reconstructed human:  - 20,000 – 200,000 triangles | - Texture maps: 8 – 12 bits per color component  - Color per vertex | - Multiple clusters/groups of meshes (e.g., different players) |

#### 4.3.5.2 Production and Capturing Systems

Dynamic meshes cannot be directly captured through 3D scanning devices. Instead, meshes can be generated either manually by artists or automatically through 3D generation algorithms. The current production methods include:

- **Manual Creation**: artists use 3D modeling software packages (such as Blender TM, Maya TM, and etc.) to manually create dynamic meshes. The artist-created meshes capture not only the external appearance of objects but also their intrinsic properties and construction details through mesh topology. High-quality meshes used in games and movies are almost exclusively created by artists.

- **Volumetric Capture Studio:** An array of multi-camera or multi-stereo cameras is placed around a recording space to capture a subject within that space. After the capture, a generation and production process is required to create the dynamic meshes. For further details, refer to Clause 4.6.7 of TR 26.928 [26928].

- **Converted from other formats:** A mesh can be extracted algorithmically from other beyond 2D representations, such as 3D Gaussians, neural fields, voxels and point clouds.

- **AI-generated meshes:** An emerging line of research generates 3D meshes in a data-driven fashion by learning from artist-created meshes using machine learning algorithms. For example, PolyGen (<https://polygen.io/>), MeshGPT (<https://nihalsid.github.io/mesh-gpt/>), MeshAnything (v1:<https://buaacyw.github.io/mesh-anything/> and v2:<https://buaacyw.github.io/meshanything-v2/>), and MeshXL (<https://meshxl.github.io/>). These methods show significant promise, particularly in terms of automating 3D asset creation. However, they are still limited by scalability, with the best method handling up to approximately 1.6K faces, and the resulting meshes often exhibit a significant quality gap compared to those crafted by artists.

#### 4.3.5.3 Rendering and Display Systems

Dynamic meshes can be rendered directly on GPUs that are highly optimized for mesh-based rendering. The following are the rendering APIs and engines for dynamic mesh processing:

- Low-Level rendering APIs:

- OpenGL: <https://learnopengl.com/Model-Loading/Mesh>

- DirectX 12: <https://microsoft.github.io/DirectX-Specs/d3d/MeshShader.html>

- Vulkan: <https://docs.vulkan.org/spec/latest/chapters/VK_NV_mesh_shader/mesh.html>

- Graphic Engines:

- Unity TM : <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Mesh.MarkDynamic.html>

- Unreal Engine TM: <https://dev.epicgames.com/documentation/en-us/unreal-engine/BlueprintAPI/DynamicMesh>

- NVIDIA RTX / OptiX TM: <https://developer.nvidia.com/rtx/ray-tracing/optix>

- Web-Based rendering APIs:

- WebGL:<https://www.khronos.org/webgl/>

- WebGPU: [https://webgpu.github.io/webgpu-samples/?sample=skinnedMesh](%20https:/webgpu.github.io/webgpu-samples/?sample=skinnedMesh)

- High-level APIs: high-level libraries use WebGL and WebGPU underneath to provide an easy-to-use, lig-htweight, and cross-browser solution for general-purpose 3D rendering. For example, Three.js (<https://threejs.org/docs/api/en/objects/Mesh.html>); Babylon.js (https://doc.babylonjs.com/features/featuresDeepDive/mesh).

Rendering can be on:

- a device for 2D presentation such as a phone

- a device for 3D presentation such as an autostereoscopic display, providing depth perception for dynamic meshes

- a device for 6DoF presentation such as VR/AR devices like Meta Quest TM, HTC Vive TM, and Apple Vision ProTM support real-time rendering of dynamic content.

#### 4.3.5.4 Supporting Information

##### 4.3.5.4.1 Test and reference sequences

<This section will be provided in a separate proposal>

##### 4.3.5.4.2 Uncompressed data size

The uncompressed data size of dynamic meshes depends on several factors, including vertex count, attribute information, level of detail (LOD), animation/deformation data and etc. A dynamic mesh sequence may require a large amount of data since it may consist of a significant amount of information changing in time. The size of dynamic mesh sequences typically ranges from a few gigabytes to several dozen gigabytes. For example, the *basketball\_player* sequence proposed by Owlii Inc. contains around 40K triangles, with texture maps at a resolution of 2048 x 2048. With a total of 600 frames, its raw data size is about 45.2GB.

##### 4.3.5.4.3 Known compression technologies

Existing compression technologies for dynamic meshes include:

- Googles’ Draco [D5], a C++ compression library designed for static meshes. Dynamic meshes are typically encoded as independent frame and the temporal coherence and redundancies in dynamic meshes are not leveraged.

- Mesh compression standards such as IC, MESHGRID, and FAMC [DM-4], previously developed by MPEGcan only compress dynamic mesh sequences with constant topological information (same vertex counts and face connections). These method can't handle dynamic meshes with time-varying topology, geometry and attribute information.

- V-DMC [DM-5], a new mesh compression standard to directly handle dynamic meshes with time varying connectivity information and optionally time varying attribute maps.The initial V-DMC test model was released by MPEG in July 2022, it is currently under Draft International Standard (DIS) status, and has been submitted to ISO with the reference ISO/IEC 23090-29. The V-DMC can be streamed in real-time using RTP-based transport [DM-6].

##### 4.3.5.4.4 Conversion from other formats

Dynamic meshes can be converted from point clouds as defined in clause 4.3.3 or voxels using software like MeshLab (https://github.com/cnr-isti-vclab/meshlab), CloudCompare (https://github.com/CloudCompare/CloudCompare), or Autodesk (https://github.com/Autodesk). Such transformation is lossy.

##### 4.3.5.4.5 Typical quality criteria

<This section will be provided in a separate proposal >

#### 4.3.2.7 Benefits and Limitations

##### 4.3.2.7.1 Benefits

The dynamic mesh format has the following benefits:

- Good visual quality. Meshes define the object's shape and structure in a fairly realistic way, allowing for finer details and realistic shading and rendering through texture mapping, making 3D assets look photorealistic.

- Most important and widely used representation for 3D assets in the commercial market. De facto standard in the film, design and gaming industries.

- Natively supported by virtually all 3D software and graphic hardware.

- Friendly to GPU, can be used for real-time rendering.

- Backward-compatible rendering. The content can be rendered on 2D displays.

##### 4.3.2.7.2 Limitations

A dynamic mesh sequence may require a large amount of data since it may consist of a significant amount of information changing in time.Standardized interoperable efficient compression, storage, and transmission of dynamic meshes have being specified. .