**3GPP TSG SA WG4#129-e S4-241519**

**Online, 19th - 23rd August 2024**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.956** | **CR** | **pseudo** | **rev** | **-** | **Current version:** | **0.0.3** |  |
|  | | | | | | | | |
| *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 Format - 3D Gaussian Splatting (3DGS) | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | China Mobile Com. Corporation | | | | | | | | | |
| ***Source to TSG:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_Beyond2D | | | | |  | ***Date:*** | | | 2024-08-08 |
|  |  | | | |  | |  | | |  |
| ***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).   During SA4#128, several scenarios were defined, that are considered to address the distribution scenarios and evaluation frameworks. However, some of the scenarios already assume a specific Representation Format that seems to be of less relevance initially.  The evaluation framework is important, once Representation formats are defined. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | This document focuses on 3D Gaussian Splatting. It is a starting point. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 4.3 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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 ===== (add to References)

[1] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Trans. Graph. 42, 4, Article 139 (August 2023), 14 pages. https://doi.org/10.1145/3592433

[2] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, “3d gaussian splatting as new era: A survey,” IEEE Transactions on Visualization and Computer Graphics, 2024.

[3] Özyeşil, Onur, et al. "A survey of structure from motion\*." Acta Numerica 26 (2017): 305-364.

[4] T. Wu, Y.-J. Yuan, L.-X. Zhang, J. Yang, Y.-P. Cao, L.-Q. Yan, and L. Gao, “Recent advances in 3d gaussian splatting,” Computational Visual Media, pp. 1–30, 2024.

[5] X. Lei, M. Wang, W. Zhou, and H. Li, “Gaussnav: Gaussian splatting for visual navigation,” arXiv preprint arXiv:2403.11625, 2024.

[6] 3DGS.zip: A survey on 3D Gaussian Splatting Compression Methods, https://3dgs.zip/

[7] Dalal, Anurag et al. “Gaussian Splatting: 3D Reconstruction and Novel View Synthesis: A Review.” IEEE Access 12 (2024): 96797-96820..

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

### 4.3.X Future formats

Editor’s Note: formats in that section will not be part of the evaluation framework of release 19, due to their maturity status, or complexity.

#### 4.3.X.1 3D Gaussian Splatting

##### 4.3.X.1.1 Definition

3D Gaussian Splatting (3DGS), also referred as Gaussian Splatting Radiance Filed, is an explicit radiance field based 3D representation that represents 3D scene or objects using a large number of 3D anisotropic balls or particles, each modeled using a 3D Gaussian distribution [1]. Each anisotropic ball or particle has a set of parameters as following:

- **Position (x, y, z):** Determine the 3D location of the center of the splat.

- **Scale (3x3 matrix):** Dictates the size and scale of the splat, affecting its shape and size.

- **Opacity (α) :** Controls the transparency, and the amount of influence the splat has on the rendered image.

- **Color parameters:** Decides the hue of the splat, adding to the visual richness. Either 3 values for (R, G, B) or Spherical harmonies coefficients.

- **Material properties:** Additional properties such as shininess, reflection, and refraction.

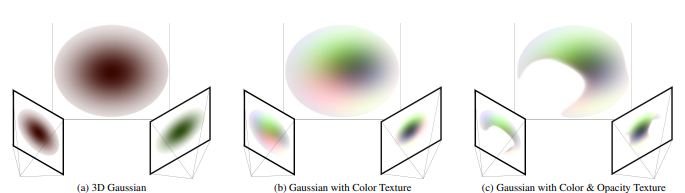


Figure 4.3.X.1-1 3D Gaussian Splatting (3DGS) representation

Pipeline and Related Technologies of 3D Gaussian Splatting [2] :

- **Structure from motion**: This process starts by creating a point cloud from images e.g., using the SFM method [3] with the COLMAP library

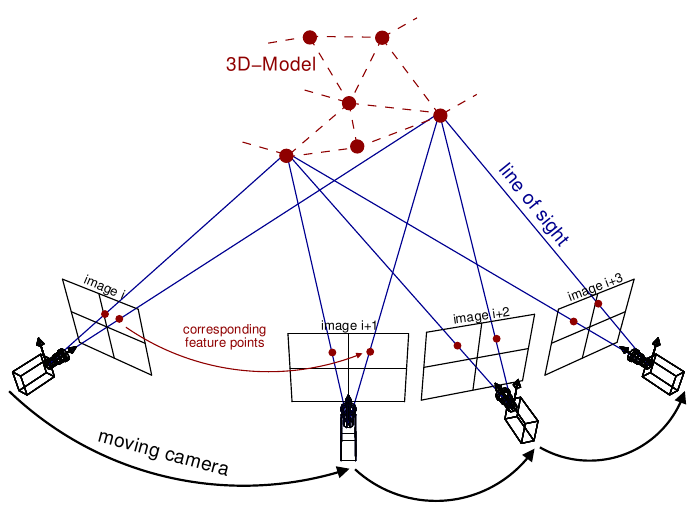


Figure 4.3.x.1-2 Structure from Motion (SfM) photogrammetric principle. Source: Theia-sfm.org (2016).

- **Convert to gaussian splats**: Each point is then converted to a Gaussian splat, which is described by parameters such as position, covariance, color, and transparency.

- **Differentiable Gaussian rasterization:** The rendering approach focuses on achieving fast overall rendering and sorting for efficient -blending without having strict limits on the number of splats. In short, the process involves sorting of gaussians by depth and grouping the tiles.

- **Training**: This process involves thousands of iterations of calculations to determine additional details about how the point could be stretched/scaled (covariance) and its opacity using Stochastic Gradient Descent (SGD). Through this process, a model is created with millions of points containing data such as position, color, covariance, and opacity.

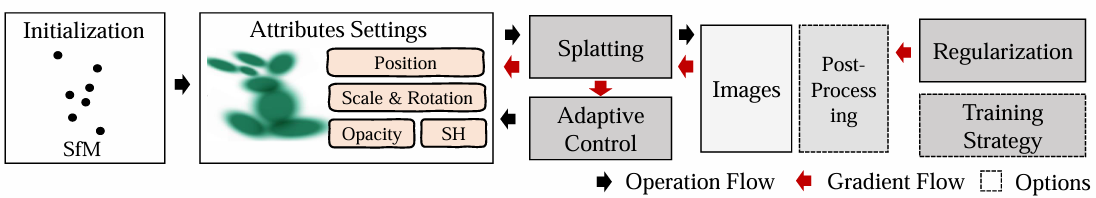


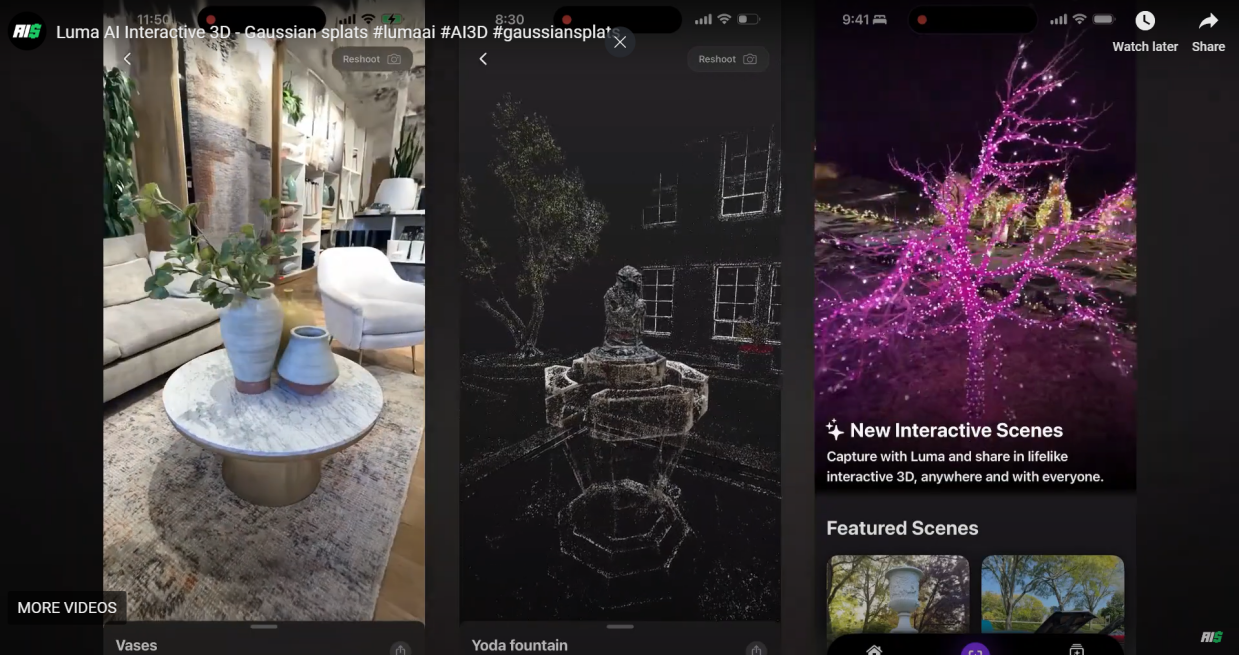
Figure 4.3.X.1-3 Pipeline and Related Technologies of 3D Gaussian Splatting

#### 4.3.X.1.2 Production and Capturing Systems

The formats as defined in clause 4.3.X.1 may be captured by mobile devices with several mobile apps (both Android and iOS devices) or online services utilize 3DGS technology. Popular options include KIRI Engine (https://www.kiriengine.app/Home), Polycam (https://poly.cam/), and Luma AI (https://lumalabs.ai/).

However, smartphones often apply automatic enhancement to each captured images, aiming for the best result. These optimizations, such as sharpening edges, adjusting exposure, and optimizing colors, can introduce noise or errors that are not suitable for Gaussian Splatting training. To mitigate these challenges, a photogrammetry capture method with professional-grade cameras (e.g., Interchangeable Lens Camera), standardized settings, and a streamlined capture process can be utilized to meet the requirements.

[Option 1: mobile devices with Luma AI:

]

Option 2: a photogrammetry capture method - https://medium.com/@heyulei/capture-images-for-gaussian-splatting-81d081bbc826

A project tutorial on capturing images for Gaussian Splatting utilized a Sony 6100 with a 16–50mm kit lens and specific camera settings. The RAW images produced had a resolution of 6024 x 4024 pixels. During Gaussian Splatting training, inputs were automatically rescaled to 1.6k. The output image size was set to 2000 x 1335 pixels, which is close to 1.6k and slightly larger.]

#### 4.3.X.1.3 Rendering and Display Systems

During the rendering process, each pixel is analogous to the discrete form of the volumetric rendering used in neural fields, enabling 3D Gaussian Splatting (3DGS) to construct complex scene appearances and achieve high-quality rendering. To facilitate high-frame-rate, high-resolution differentiable rendering, 3DGS employs a tile-based rasterizer [4].

This rasterizer first divides the image into 16 × 16 tile, assigning an index to each tile. For each Gaussian primitive, the rasterizer determines which tiles the primitive’s projection intersects and generates a key-value pair for each intersecting tile. The key is 64-bit, with the upper 32 bits representing the tile index and the lower 32 bits representing the projection depth of the Gaussian primitive. By constructing these key-value pairs, the rasterizer only needs to perform a global sort on all pairs, eliminating the need for additional sorting of primitives for each pixel.

After sorting, the key-value pairs derived from each tile are stored in contiguous memory intervals. The rendering process for each tile is then managed by a CUDA thread block, with the number of threads within each block matching the number of pixels in the tile. Each thread is responsible for the alpha blending process for its corresponding pixel, completing the final rendering.

[commercial services:

* Gaussian Splatting 3D Creator, Viewer & Editor: <https://poly.cam/tools/gaussian-splatting>
* For Unity, the Gaussian Splatting package (Unity Asset Store) and for Unreal Engine, the UE GaussianSplatting plugin (Unreal Engine Marketplace) are examples of such developments.
* Another solution was implemented for the collaborative 3D platform Spline , which showcased their support for 3D Gaussian splatting with a very exciting demo, discussed further here: https://designmusketeer.com/spline-new-update-gaussian-splatting.]

#### 4.3.X.4 Supporting Information

- Typical quality criteria for evaluating the format

- In the image domain: The Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM),Learned Perceptual Image Patch Similarity (LPIPS), the resultant size in megabytes (MB), the training time, and required storage.

- In the 3D point domain: point to point/ point to plane, / point to surface for position. Attribute PSNR.

- Conversion from other formats (lossless, lossy)

- Point Clouds: Converting point cloud representations to 3DGS can effectively fill in the point clouds’ holes, which is typically done after high-precision reconstruction of the point clouds. Conversely, 3DGS can also be converted into point clouds, followed by voxelizing the point cloud into 3D voxels and then projecting them onto 2D BEV grids [5].

NOTE: filling holes in point cloud, can also be achieved by different techniques.

- Mesh: A considerable amount of work has discussed how to convert 3DGS to Mesh. Once converted, they can be further optimized to achieve better geometric and appearance,

- Uncompressed data size

A Gaussian splat of a scene is a representation of 3D points. On average, a splat contains between 0.5 and 5 million of these 3D points. Each 3D point has unique parameters that represent the scene as accurately as possible.

- Known compression technologies:

Early research on 3DGS compression is ongoing. Vector quantization (VQ) [6] is widely applied in current research for 3DGS compression and encoding. It first categorizes data into clusters of similar vectors and then represents them via a compact codebook. MPEG has started an exploration that is looking at the most appropriate representation formats for 3DGS and various coding strategies.

- Extensibility of the format

The format is undergoing massive academic and industrial research, can be further expanded for more capabilities and is hence not stable. Example of use cases include: 3DGS can be further expanded for more capabilities, including dynamic 3DGS [4, 7], surface representation from 3DGS [4, 7], editable 3DGS [4, 7], 3DGS with semantic understanding [4, 7], and 3DGS-based physics simulation [4, 7].

##### 4.3.X.1.5 Benefits and Limitations

###### 4.3.X.1.5.1 Benefits

- Real-time Rendering with GPU acceleration.

- Accurate Reconstruction

- Simple and explicit representation

- Ability to render complex scenes in real-time

- Interpretability of the representation

###### 4.3.X.1.5.2 Limitations

- There is a lack of industry agreement on the 3DGS format(s), due tono stable representation and compression format exists for static and dynamic 3DGS.

- Static and Dynamic 3DGS formats is evolving, multiple options are considered in current academic and industrial research. For dynamic, such research include modeling in 4 dimensions (i.e. temporal), time evolving 3DGS, and MLP predicted motion for 3DGS among others.

- High memory usage

- Not yet fully compatible with existing rendering pipelines