3D Gaussian Splatting Methods for Real-World Scenarios

Ivana Petrovska* and Boris Jutzi
Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany -
ivana.petrovska@partner.kit.edu, boris.jutzi@kit.edu

KEY WORDS: 3D Gaussian Splatting, Neural Radiance Fields, Multi-View Stereo, 3D Reconstruction, Point Cloud Comparison.

ABSTRACT:

3D Gaussian Splatting (3DGS) is an innovative solution for explicit point-based 3D representation where each point is represented as a Gaussian distribution. Using calibrated images and sparse point cloud for initialization, the scene is reconstructed by optimizing the Gaussian position, orientation, shape and appearance. In this contribution, we present a comprehensive overview of 3DGS methods available on complimentary radiance field reconstruction platforms namely, the original 3DGS implementation as reference, Splatfacto, 3DGS-MCMC and 3DGS-LumaAI to address a broader audience in industry and non-technical users as well. Being an indispensable part of our environment, we are particularly interested in vegetation since the irregular and complex shape of plants and trees, especially dense foliage can challenge the 3D reconstruction. We evaluate the geometric accuracy and completeness in two real-world scenarios, one occlusion-free indoor and one outdoor scenario where the object of interest is placed behind vegetation to investigate how the methods can reconstruct the underlying geometry behind occlusion. To investigate if 3DGS methods can challenge traditional and state-of-the-art 3D reconstruction approaches we compare the results with Multi-View Stereo (MVS) and Neural Radiance Fields (NeRFs). The evaluation is based on point cloud comparison against a ground truth mesh. Just behind MVS, the original 3DGS implementation achieves second best accuracy results outperforming NeRFs in both scenarios, making it the most accurate 3DGS method. 3DGS-MCMC achieves the best and third best completeness for each scenario respectively, making it competitive with MVS and NeRFs in real-world setting. Moreover, we demonstrate 3DGS ability to reliably reconstruct the geometry behind vegetation occlusion indicating the potential for large-scale forestry applications, allowing canopy reconstruction, biomass estimation and agricultural monitoring.

1. INTRODUCTION

In recent years, the field of computer vision has witnessed remarkable advancements in 3D reconstruction from multiple calibrated views compared to the traditional Multi-View Stereo (MVS) algorithm which relies on cross-view correspondence matching and triangulation to estimate per-pixel depth value. Innovations such as Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021) have achieved substantial breakthrough in implicit scene representation by leveraging neural networks to represent a scene as a continuous 5D function that takes as input 3D spatial coordinates and 2D viewing direction and outputs the density and color at that spatial location. Despite these advancements, NeRF-based methods still contend with challenges such as computationally expensive optimization due to the significant cost of ray tracing and uninformative data for storing points in empty space. Moreover, the geometry is only learned implicitly, encoded in the density field which is a continuous function and carries uncertainty (Goli et al., 2024, Jäger et al., 2023) leading to incomplete and noisy surface representation (Petrovska et al., 2023, Oechsle et al., 2021).

After NeRFs, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has become an innovative solution for explicit point-based 3D reconstruction where each point is represented as a Gaussian distribution. Incorporating images with corresponding poses initialized on a sparse point cloud usually obtained by Stucture-from-Motion (SfM) or depth maps (Jäger et al., 2024), the scene is captured by optimizing the Gaussian position, orientation, shape and color represented as spherical harmonics (SH). While 3DGS does not require dense point sampling for each ray, it does need a substantial number of Gaussians to maintain a high level of detail. Optimizing the Gaussian primitives is performed through gradient descent, thus the Gaussians with large gradients are densified by cloning or splitting. Unlike ray-based approaches which sample points for each pixel from training views, 3DGS optimizes on full images requiring different training formation and doesn’t compute points in empty space.

* Corresponding author

In this contribution, we present a comprehensive overview of 3DGS methods available on complimentary 3D reconstruction platforms employing radiance fields, which are highly valuable due to their efficiency, while allowing reconstruction with less technical competence, to address a broader audience in industry and non-technical users as well. We qualitatively and quantitatively evaluate the geometric accuracy and completeness to offer deeper insights into the practical applications and limitations in real-world scenarios. We are particularly interested in vegetation since the irregular and complex shape of plants and trees, especially dense foliage can challenge the overall accuracy and completeness of the reconstructed scene. Our investigations are based on one occlusion-free indoor scenario and one outdoor scenario where the object of interest is placed behind vegetation to investigate how the methods can reconstruct the underlying geometry behind occlusion. The evaluation is based on point cloud comparison against a ground truth mesh, since point clouds are a main 3D representation in computer vision and photogrammetry. To investigate if 3DGS methods can challenge traditional and state-of-the-art 3D reconstruction approaches we compare the results with MVS and NeRFs respectively.

In summary, our main contributions are:

- We qualitatively and quantitatively evaluate the 3D geometric accuracy and completeness of 3DGS methods available on complimentary radiance field reconstruction platforms for real-world scenarios, tackling occlusions.

- We compare the results with MVS and NeRFs, to investigate if 3DGS methods can challenge traditional and state-of-the-art 3D reconstruction approaches.
- We demonstrate that 3DGS achieves second best accuracy results outperforming NeRFs indicating a consistent performance in real-world setting.

The contribution is organized as follows. In Section 2 we give an overview of the 3D reconstruction methods, in Section 3 information about the dataset and evaluation metrics along with implementation details is provided, the qualitative and quantitative results addressing accuracy and completeness are presented in Section 4, the discussion is laid out in Section 5 and Section 6 reports the concluding insights.

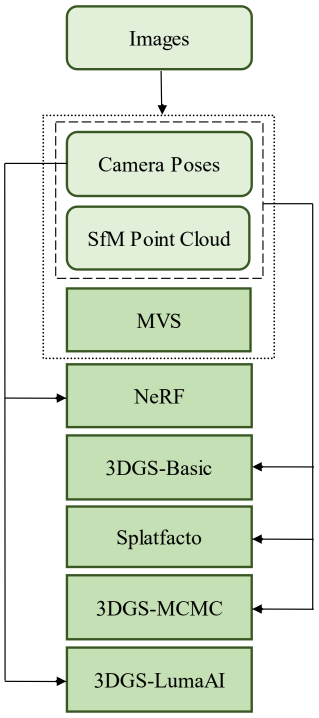

Figure 1. We estimate the camera poses and sparse point cloud through SfM, followed by dense MVS reconstruction (Section 2.1). The images along with the poses are input to NeRF (Section 2.2) and 3DGS-LumaAI (Section 2.3.4), while 3DGS-Basic (Section 2.3.1), Splatfacto (Section 2.3.2) and 3DGS-MCMC (Section 2.3.3) additionally require the sparse point cloud for initialization.

2. 3D RECONSTRUCTION METHODS

We evaluate how 3DGS methods available on radiance field reconstruction platforms can represent real-world indoor and outdoor scenarios. As a 3DGS representative we refer to the original implementation. Due to the explicit representation, the Gaussian mean is considered object geometry in the form of a point cloud for all methods. For more comprehensible insights in the accuracy and completeness, we compare the results with current 3D reconstruction methods, MVS and NeRFs (Figure 1).

2.1 Multi-View Stereo (MVS)

The camera poses along with a sparse point cloud are estimated through the incremental SfM pipeline (Schonberger and Frahm, 2016) using SIFT (Lowe, 2004) for feature extraction and matching. Based on the SfM output, the dense point cloud is computed from the matching points using multi-view triangulation that effectively captures the spatial structure of the scene through MVS (Schönberger et al., 2016). The depth and normal information for every pixel is computed by analyzing pixel correspondences across the images. Fusion of the depth and normal maps of multiple images in 3D produces a dense point cloud of the scene.

2.2 Neural Radiance Fields (NeRFs)

NeRFs approximate a 5D plenoptic function with a learnable multi-layer perceptron (MLP), whose input is a 3D location $x = (x, y, z)$ and 2D viewing direction (θ, ϕ) and whose output is volume density σ and emitted color $c = (r, g, b)$ at that spatial location. As NeRF representative we employ Nerfacto (Tancik et al., 2023), which is a recommended method for real-world data of static scenes. Although the core of the method is heavily influenced by Mip-NeRF360 (Barron et al., 2022), it combines several components from different NeRF methods (Wang et al., 2021, Müller et al., 2022, Martin-Brualla et al., 2021, Verbin et al., 2022) to achieve fast training and high reconstruction quality. The method uses a small neural network with hash encoding for computational efficiency, while achieving comparable accuracy. To produce the initial set of samples of the scene, a piece-wise sampler is employed. It allocates half of the samples uniformly up to a fixed distance from the camera, followed by samples that are distributed such that the step size increases with each sample, to achieve a dense set of samples for nearby objects. These samples are then fed into a proposal sampler, which consolidates the sample locations into regions of the scene that contribute most to the final render, typically the first surface intersection with the objective to learn a guidance for the NeRF ray sampling near the surface of objects within the scene. For real-world unbounded scenes, a scene contraction which compresses the infinite space into a fixed-size bounding box is used. The point cloud is extracted by back-projecting points from the depth maps and mapping to 3D coordinates with respect to the camera poses for each input image. Per-pixel depth is derived from the expected ray termination in the density field and calculated by accessing the median density response on the sampled ray.

2.3 3D Gaussian Splatting (3DGS)

In the following, we present the evaluated 3DGS methods, specifically 3DGS-Basic, Splatfacto, 3DGS-MCMC and 3DGS-LumaAI.

2.3.1 3DGS-Basic. We refer to the original implementation as 3DGS-Basic. The scene is represented with many differentiable 3D Gaussian primitives parameterized by position centered in the Gaussian mean $\mu \in \mathbb{R}^3$, covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ decomposed into a scaling vector $s \in \mathbb{R}^3$ and a rotation quaternion $q \in \mathbb{R}^4$, opacity $\alpha \in \mathbb{R}$ and color $c \in \mathbb{R}^3$ represented via SH coefficients. The scene is initialized on a sparse point cloud where the initial covariance matrices have axes equal to the average distance of a point to its closest three neighbors, meaning that initially the Gaussians are spheres. During training the parameters of the Gaussians are optimized through gradient descent by many rendering iterations to best fit the training dataset. The algorithm progressively densifies the scene by removing, splitting and duplicating the Gaussians at fixed iterations based on the adaptive density control. This leads to perpetually growing the number of 3D Gaussians over training time, resulting in a denser point cloud than the initial sparse point cloud. The 3D Gaussians are trained to minimize the photometric loss $\mathcal{L}$, where $\mathcal{L}_1$ is norm of the per pixel color difference combined with D-SSIM term and λ=0.2:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{1}$$

For visualization purposes, we also color code the point cloud by converting the SH back to RGB values.

2.3.2 Splatfacto. The CUDA-based backbone of Splatfacto is the gsplat library (Ye et al., 2024) which includes optimization improvements for speed, memory and convergence times compared to the original 3DGS-Basic implementation. It features a front-end with Python bindings and a back-end with highly optimized CUDA kernels.

2.3.3 3DGS-MCMC. Based on the principles of Markov Chain Monte Carlo sampling, 3DGS-MCMC (Kheradmand et al., 2024) reinterprets the placement and adjustment of 3D Gaussians as a process of sampling from a probability distribution. Instead of using heuristics for cloning and splitting the Gaussians, it employs a relocalization scheme to move low opacity Gaussians to the locations of Gaussians with high opacity. This removes unused Gaussians, thus managing their number effectively, thereby reducing the computational overhead and required amount of memory.

2.3.4 3DGS-LumaAI. We use the 3DGS method that it's a proprietary iteration of the original implementation. The goal is to hybridize the performance gain of real-time with robust cloud-based rendering. Build upon luma-web library for immersive environments, it includes LumaSplatsWebGL, which is a WebGL-only Gaussian Splatting implementation designed to be integrated with 3D frameworks and LumaSplatsThree, which is a Three.js implementation that uses LumaSplatsWebGL under the hood.

3. EXPERIMENTS

After a brief description of the dataset in Section 3.1, we outline the evaluation metrics in Section 3.2, followed by the implementation details for each method in Section 3.3.

3.1 Dataset

Our investigations are based on two real-world scenarios from the STELLA[1] dataset (Petrovska and Jutzi, 2024), namely *Original* which is occlusion-free and *Vegetation* depicting vegetation as non-transparent occlusion (Figure 2). The object whose geometry ought to be evaluated is 0.7m tall Buddha statue (further on referred to as object) placed on a rectangular plate. Each scenario consists of 125 high resolution images captured on a circular trajectory around the object. Due to memory limitations and efficiency the images are downsampled to 1840x1228 pixels and converted into .png format for lossless compression. As ground truth, we use a mesh generated using Structured Light Imaging (SLI) with 0.1mm accuracy.

[1] `https://github.com/sqirrel3/STELLA`

3.2 3D Evaluation

We remove redundant data unrelated to the evaluation and keep just the object. All point clouds are aligned in the same metric space as the ground truth mesh for evaluation. After manual coarse alignment, Iterative Closest Point (ICP) (Besl and McKay, 1992) which finds an optimal rigid transformation to align two point sets is used. Subsequently, we report qualitative and quantitative accuracy and completeness of the reconstructed point clouds (Petrovska and Jutzi, 2025).

Accuracy. Accuracy quantifies how closely the reconstructed point cloud represents ground truth locations. We use cloud-to-mesh which computes the displacements between each point in the compared point cloud and the nearest facet in the reference mesh through Euclidean distance. The geometric distortions are estimated by: Mean Error (Mean), Standard Deviation (SD) and Root Mean Squared Error (RMSE) in CloudComapre[2].

Completeness. Completeness (*Cpl*) measures to what extent the ground truth surface is covered and is calculated as the ratio of the number of covered points to the total number of points in the reference cloud. A higher percentage indicates higher completeness, conditioned by the number of points (*Npts*). All points in the ground truth within the threshold of 5mm distance of an estimated point contribute to the completeness. For this purpose we converted the mesh into a point cloud by subsampling 10M points on the mesh and perform the calculation in our python script.

3.3 Implementation Details

MVS. In COLMAP[3] we use the exhaustive matching strategy in the SfM phase which involves comparing features between every possible pair of images to find matching points. The images with corresponding poses are input to NeRF and 3DGS-LumaAI, while 3DGS-Basic, Splatfacto and 3DGS-MCMC additionally require the sparse SfM point cloud for initialization.

NeRF. We use the default parameters for Nerfacto in the Python-based framework Nerfstudio[4] v1.1.0. First 256 samples from the piecewise sampler are generated, which get resampled into 96 samples in the first iteration of the proposal sampler, followed by 48 samples in the second. For comparability, the pose refinement is disabled.

3DGS-Basic. The training is performed on the default parameters with learning rates of 0.0025 for SH, 0.05 for opacity adjustments, 0.005 for scaling operations, 0.001 for rotation transformations and 0.0002 threshold of positional gradient norm for densifying the Gaussians. The densification starts from the 500th iteration until 15.000 and the Gaussians are densified every 100 iteration. The opacity is set close to zero every 3.000 iterations to prevent the method from getting stuck with floaters close to the camera poses.

Splatfacto. For Splatfacto, also released in the scope of Nerfstudio, we use the default hyperparameters which correspond to 3DGS-Basic, only the limit that decides if points should be densified based on the positional gradient is 0.0008.

3DGS-MCMC. Being the recommended reconstruction method for most scenes, we use the 3DGS-MCMC implementation in

[2] `https://github.com/CloudCompare/CloudCompare`
[3] `https://github.com/colmap/colmap`
[4] `https://github.com/nerfstudio-project/nerfstudio`

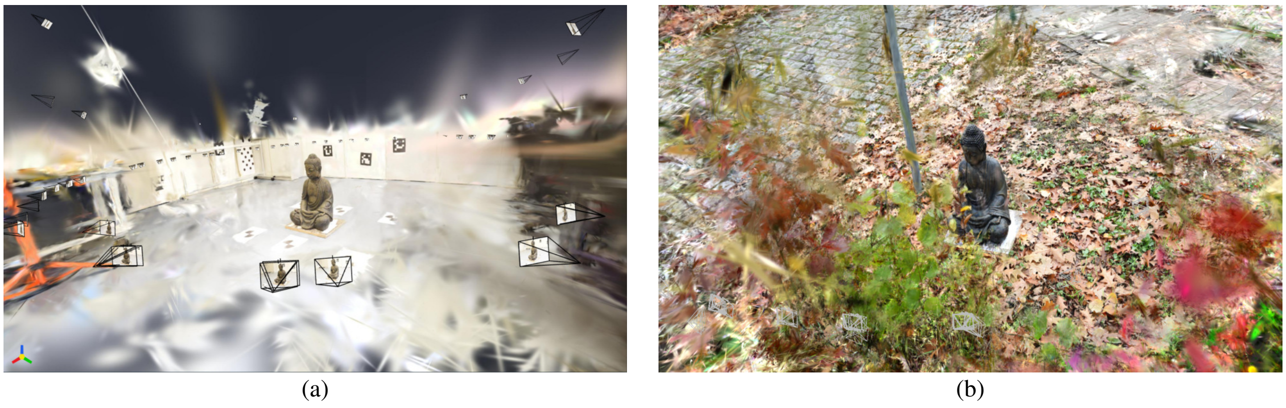

Figure 2. (a) *Original* reconstructed by Splatfacto and (b) *Vegetation* through 3DGS-MCMC reconstruction. The Gaussian splats which act like blobs in space with view-dependent appearance are visualized along with the camera poses.

Postshot[5] v0.4.0, which offers an end-to-end solution for training and rendering radiance fields within a seamless workflow. We use the recommended parameters, with maximum splat count of 3M. This specifies the number of splat primitives that training will at most create in the scene and directly affects the required memory and disk space as well as the level of fine detail.

3DGS-LumaAI. In the Interactive Scenes space released by Luma Labs AI[6] we upload the images with corresponding poses and choose the camera type - normal in our case, which are the only adjustable parameters. The 3D reconstruction is done remotely in the cloud.

The training for NeRF, 3DGS-Basic, Splatfacto, as well as 3DGS-MCMC incorporates 30.000 iterations and every 8th frame is taken for test during training, except for 3DGS-MCMC in Postshot where this parameter cannot be adjusted and all images are taken for training. For 3DGS-LumaAI we used the default settings, whose access is not available. To convert the SH back to RGB for the point clouds generated from Splatfacto, 3DGS-MCMC and 3DGS-LumaAI we use an open source library[7]. All experiments regarding the pose estimation, training (except 3DGS-LumaAI which is web-based) and evaluation are performed on a desktop PC with an Intel i9 CPU with 32GB RAM and Nvidia RTX3090 GPU.

4. RESULTS

We report qualitative, where the geometric reconstruction as point clouds are visualized (Figure 3) as well as the cloud-to-mesh errors (Figure 4) and quantitative results to numerically present the accuracy and completeness for each method separately (Table 1).

MVS. MVS reliably reconstructs the object and achieves a sharp result in both scenarios. However, in *Original* we can observe black and white artifacts, especially noticeable on the spikes on the head and left shoulder. Moreover, noise in the lap is present due to lack of image information. These points have high error values. Overall, it demonstrates superior accuracy among the methods in both scenarios with lowest error displacements, thus achieving the highest correspondence with the ground truth with RMSE of 1.43mm and 3.23mm respectively. This is complemented by high completeness of 97.35% with a substantial number of points, implying a well-balanced trade-off between precision and surface coverage. Although it experiences a decrease in completeness to 75.95% in *Vegetation*, due to the gaps in the geometry of the occluded parts, it still outperforms other methods and exhibits the best results.

NeRF. NeRF point clouds show realistic representation and can capture the overall spatial arrangement of the scene completely while being able to capture complex geometric details, e.g. the spikes on the head, even behind occlusions. The lap is also reconstructed with more points compared to MVS. However, triggered by the challenges of occlusions in *Vegetation* the reconstruction is fuzzy and has noise on the outer object surface, but the occluded parts are better reconstructed compared to all other methods. Hence, it achieves second best results in both scenarios regarding completeness suggesting its strength in generating detailed 3D reconstruction albeit with accuracy trade-off where it performs considerably worse, with high RMSE of 5.36mm and 12.67mm accordingly. This is mostly caused by the points inside the object which have the highest error values.

3DGS-Basic. 3DGS-Basic shows better object reconstruction compared to Splatfacto and 3DGS-LumaAI. Nonetheless, the point clouds are very unstructured with uneven spread of points; complex geometry and edges are represented with a significant number of points, while the homogeneous areas are sparse. We argue that this is because the point cloud densification is guided from the input sparse point cloud from SfM which is able to identify only distinctive points in the feature extraction and matching phase. Generally, 3DGS-Basic achieves second best accuracy results in both scenarios and exhibits highest accuracy results among the evaluated 3DGS methods. However, it suffers from lower completeness with 82.75% for *Original* and 42.71% for *Vegetation*. It is worth noting that the completeness is almost twice as lower for *Vegetation* making it sensitive to occlusions, which applies to all 3DGS methods.

Splatfacto. In *Original*, Splatfacto reliably reconstructs the object depicting complex texture - the spikes on the head, edges - lines of clothes and homogeneous object parts - knees. However, the point cloud is very sparse for *Vegetation* which indicates difficulties in managing the complexities of occlusions in unbounded scenes. Nevertheless, the complex geometry on the head is still reliably reconstructed. Splatfacto shows poor accuracy and large RMSE values of 7.28mm and 13.80mm, combined with relatively low completeness scores, 79.83% and 31.80% for each scenario respectively.

[5] `https://www.jawset.com/`
[6] `https://lumalabs.ai/interactive-scenes`
[7] `https://github.com/francescofugazzi/3dgsconverter`

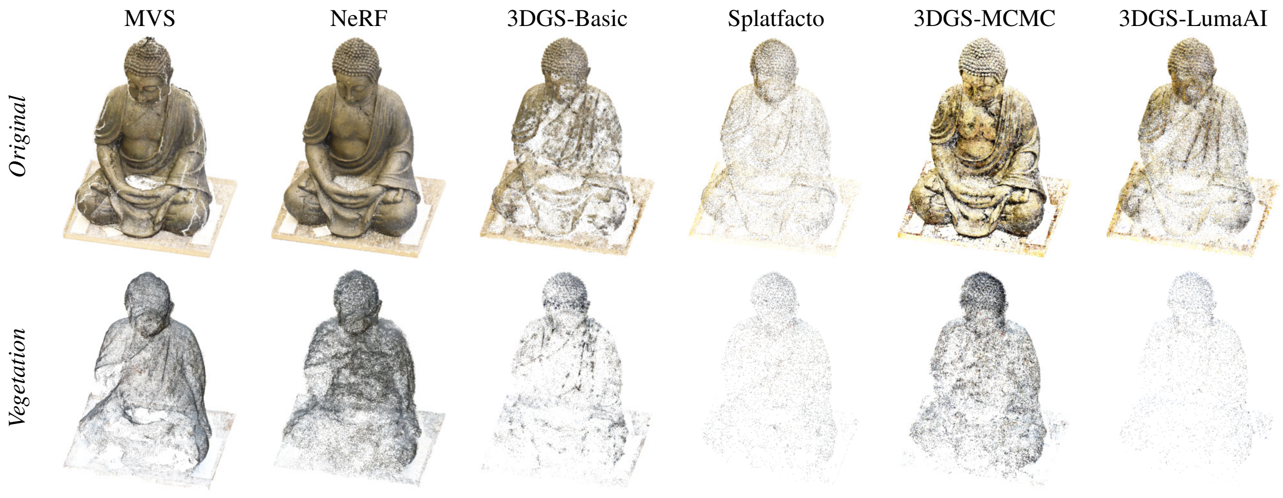

Figure 3. Point cloud geometric reconstruction for all evaluated 3D reconstruction methods in both scenarios. Due to the occlusions all point clouds in *Vegetation* have less points compared to the occlusion-free *Original* scenario. The geometry for all 3DGS methods is represented by the Gaussian mean.

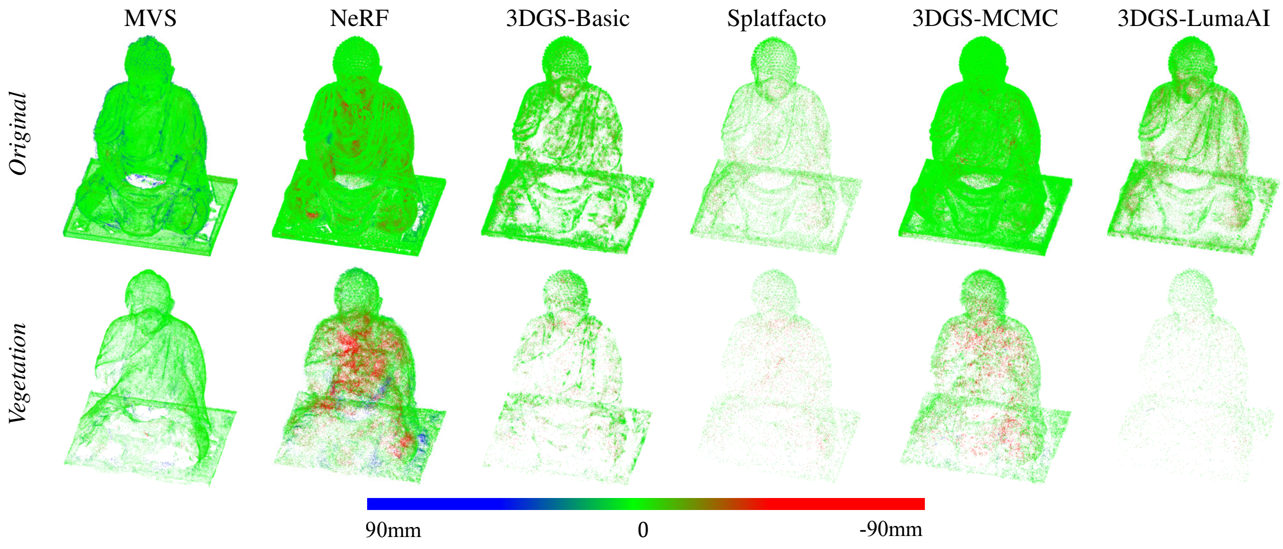

Figure 4. Geometric accuracy through cloud-to-mesh distances against the ground truth mesh for both scenarios, *Original* and *Vegetation*. The error displacements correspond to the color ramp. Taking into account the orientation of the normal vector the calculated cloud-to-mesh distance is signed; a point is considered outside the mesh when the distance is positive and inside when it is negative. Except MVS, the reconstructed point clouds have artifact points inside the object which distort the accuracy.

3DGS-MCMC. The 3DGS-MCMC method shows highly detailed object reconstruction for *Original*, outperforming all methods in completeness achieving almost absolute point coverage with 99.45% and strikes a favorable balance between point coverage and accuracy with moderate displacements below 6mm. In *Vegetation*, the intricate geometry - spikes on the head are still detailed, but the homogeneous - plate and lap are sparse. In addition, the edges - lines on the clothes are not recognizable which is in some part conditioned by the occlusions. The number of points significantly drops, but it’s still enough for third best completeness results. However, it fails to accurately reconstruct the object with highest RMSE of 17.36mm. Altogether, it reaches the highest completeness in both scenarios among the 3DGS methods.

3DGS-LumaAI. Similarly to 3DGS-Basic, the point distribution in *Original* is scattered with homogeneous parts sparsely represented. Nevertheless, the object geometry is reconstructed with completeness of just below 90% outperforming Splatfacto and 3DGS-Basic. In contrast, the completeness drops drastically in *Vegetation* to only 7.173 points and the object features are almost unrecognizable. This leads to the lowest completeness percentage of just 27.02%. When it comes to accuracy, it shows robust results with errors in the range of 7mm for both scenarios.

NeRFs and all 3DGS methods show lower accuracy than MVS because the geometry is reconstructed based on minimizing the image reconstruction loss between training and rendered images. The points don’t align tightly with the object surface and moreover, artifact points inside the object are present which distort the accuracy and don’t contribute to the completeness (Petrovska and Jutzi, 2024). They are projections from input views that weren’t moved into their correct place geometrically.

Table 1. Quantitative results of the cloud-to-mesh comparison addressing the accuracy and completeness for each 3D reconstruction method. The results depend on the chosen 3DGS method. 3DGS-Basic achieves second best accuracy results, outperforming NeRFs in both scenarios. 3DGS-MCMC exibits the highest completeness among the evaluated 3DGS methods and overall for *Original*. The first, second and third best results are highlighted.

Scenario	Method	Accuracy (mm)			Completeness (%)	
		Mean ↓	SD ↓	RMSE ↓	Npts	Cpl ↑
Original	MVS	0.16	1.42	1.43	845.456	97.35
	NeRF	-2.94	4.48	5.36	1.682.388	98.23
	3DGS-Basic	-2.02	4.29	4.74	352.298	82.75
	Splatfacto	-3.15	6.56	7.28	51.209	79.83
	3DGS-MCMC	-1.60	5.58	5.80	1.556.040	99.45
	3DGS-LumaAI	-3.47	6.11	7.02	167.470	88.77
Vegetation	MVS	0.53	3.18	3.23	117.427	75.95
	NeRF	-5.43	11.44	12.67	181.058	69.95
	3DGS-Basic	-2.54	7.05	7.49	40.807	42.71
	Splatfacto	-6.21	12.33	13.80	10.518	31.80
	3DGS-MCMC	-7.35	15.70	17.36	69.719	65.93
	3DGS-LumaAI	-1.95	7.32	7.58	7.173	27.02

The error displacements tend to increase faster to the negative because those are the points behind the mesh surface and inside the object. The errors with positive values lie above the nearest mesh triangle and thus most likely represent noise and outlier points. Overall, the 3DGS-Basic achieves second best accuracy results, outperforming NeRFs in both scenarios, making it the most accurate 3DGS method. Among the evaluated 3DGS methods, 3DGS-MCMC exhibits highest completeness competitive with MVS and NeRFs. Without affecting the geometry, we can observe color differences in the object reconstruction between the scenarios. *Original* is captured indoors under controlled lighting, which is different from *Vegetation*, captured outdoors under non-constant natural illumination.

5. DISCUSSION

In this contribution we evaluate the 3D geometry reconstructed by 3DGS methods implemented on different radiance field platforms against MVS and NeRFs, analysing the accuracy and completeness in occlusion-free and non-transparent vegetation occlusion scenario. In the following, we discuss the qualitative and quantitative results and argue their application influenced by the code base availability and training time (Table 2).

Table 2. Algorithm access and training time for the evaluated methods. The code for the 3DGS-MCMC implementation in Postshot and the 3DGS-LumaAI is not open-source, limiting their application on account of ease of use. The fastest 3D reconstruction method is NeRF, followed by Splatfacto.

Method	Open-source	Training time
MVS	✓	1h 15min
NeRF	✓	15min
3DGS-Basic	✓	49min
Splatfacto	✓	25min
3DGS-MCMC	✗	43min
3DGS-LumaAI	✗	45min

MVS. Although MVS excels in pin-point accuracy while achieving high completeness, it is the slowest 3D reconstruction method with average reconstruction time of 1h 15min, due to the costly dense matching. However, the code is open-source meaning that parameters for both SfM and MVS phase can be accessed and modified.

NeRF. Since the number of exported points can be adjusted, we chose a number close to the highest (according to 3DGS-MCMC for *Original* and MVS for *Vegetation* scenario) for a fair comparison. However, the number of points doesn't necessarily imply a higher completeness because they don't represent object surface points. Both NeRF point clouds have outer noise and contain artifact points inside the object which don't improve the completeness and decrease the accuracy. Nevertheless, NeRF achieves second best completeness results in both scenarios while being the fastest method with just 15min of training time, indicating its high efficiency. Released with an open-source code, it is a competitive option for real-world settings.

3DGS-Basic. 3DGS is a method for representing radiance fields by explicitly storing a collection of 3D volumetric Gaussians building upon the principles of MVS and NeRFs. Hence, it achieves the best accuracy among the 3DGS methods and second best overall, outperforming NeRF in both scenarios. The completeness is average due to the sparsely reconstructed homogeneous object parts, caused by the lack of points for these parts in the SfM point cloud used for initialization. The code is open for contributions, but the training time of almost 50min makes it the slowest 3DGS method. This stems from the high computational complexity caused by the large number of Gaussians, the non-linear operations for Gaussian optimization and large memory requirements for storing all Gaussians along with their parameters.

Splatfacto. Splatfacto achieves the least reliable results from the 3DGS methods and performs worse than MVS and NeRF (Haitz et al., 2024), limiting its usage in real-world scenarios. Although it matches the original 3DGS implementation for the most part, we suspect that the gradient descent threshold for densification significantly affects the results, being the only hyperparameter that differs. The cloning, splitting and removing of the Gaussians is more rigorous if this value is higher. In addition, we notice another difference from the original code: the percentage of scene extent a point must exceed to be forcibly densified, which is 0.01 by default and multiplies the scene extent. Splatfacto assumes that the scene is normalized to -1 and

1 (default setting auto scale camera poses to -1 and 1 so the scene extent is 1). This could explain the decrease in accuracy and completeness for the unbounded *Vegetation* scenario. Nonetheless, the high memory usage due to the memory fragmentation of Gaussian culling is resolved by actively emptying CUDA cache, making it almost two times faster than the rest of the 3DGS methods with 25min of training time.

3DGS-MCMC. Although it achieves the highest completeness for *Original* with moderate accuracy, for *Vegetation* which is unbounded, the point coverage drops to around 65% and it yields the highest error displacements. We left the maximum number of Gaussian primitives for both scenes to default 3M, which is probably not enough because the scene is too big for the amount of Gaussians. The code for the 3DGS-MCMC implementation in Postshot is not open-source with limited parameter adjustment, restricting more detailed insights into the results. Moreover, the evaluation strategy differs, as it doesn’t split the dataset and all available images are used for training. The training time is moderate with slightly above 40min.

3DGS-LumaAI. Although the method shows robustness and moderate accuracy for *Vegetation*, it exhibits the lowest completeness of just 27.02% which is not enough to reliably capture the object’s geometry and appearance. Moreover, we can’t be certain about the initialization and evaluation, because the user interface allows just images and camera poses to be uploaded. We believe that random initialization is used because the object reconstruction is similar to 3DGS-Basic, except that it has fewer points. The lack of technical documentation restricts its applicability even though it is faster than 3DGS-Basic with training time of 45min. On the other hand, the workflow is straightforward, without requiring technical knowledge or high-performance hardware and software since the processing takes place remotely.

In general, 3DGS methods show competitive results with MVS and NeRFs in both scenarios. However, all 3DGS methods show a higher drop in the completeness in *Vegetation* compared to MVS and NeRF, limiting their robustness in occlusion scenarios. We can also notice color differences between the 3DGS methods, which only affect appearance. In 3DGS-Basic we converted the SH back to RGB in the codebase, while for Splatfacto, 3DGS-MCMC and 3DGS-LumaAI another library is used. The accuracy and completeness depend on the chosen method and parameters. Out of the six evaluated methods, four are open-source: MVS, NeRF, 3DGS-Basic and Splatfacto. These methods offer higher customization potential while enabling fine-tuning based on specific scene requirements or optimization criteria. In contrast, the pipeline is atomized in Postshot and Luma AI, restricting parameter adjustments. This last aspect represents both an advantage, for the simplicity of the workflow doesn’t require technical knowledge, but also a limitation as it does not allow data management and possibility of intervening during the various phases of the reconstruction. These methods are more user-friendly, requiring less technical expertise, but also offer less flexibility in adapting to specific scene characteristics (Basso et al., 2024).

6. CONCLUSION

In this contribution we evaluate the accuracy and completeness of 3DGS methods available on complimentary radiance field platforms namely, the original 3DGS implementation as reference, Splatfacto, 3DGS-MCMC and 3DGS-LumaAI. We compare the reconstructed point clouds with current 3D reconstruction methods, MVS and NeRFs in two real-world scenarios: *Original* which is an occlusion-free and *Vegetation* tackling vegetation as non-transparent occlusion. 3DGS accuracy and completeness depend on the chosen method. Overall, the original 3DGS implementation achieves second best accuracy results outperforming NeRFs, while MVS is still the most accurate 3D reconstruction method. Concerning completeness, 3DGS-MCMC achieves best and third best results for each scenario respectively. Taking into account the training time, NeRF is the fastest method, followed by Splatfacto. Due to the time-consuming dense matching, MVS as expected is the slowest 3D reconstruction method.

Moreover, we demonstrate 3DGS ability to reliably reconstruct the geometry behind vegetation occlusion allowing canopy reconstruction, biomass estimation and agricultural monitoring. Beside Light Detection and Ranging (LiDAR) (Dubayah and Drake, 2000), an extended 3DGS shows immense potential for forestry applications. However, processing extensive landscapes require high memory usage and computational demands, particularly for dense and occluded forest regions.

REFERENCES

Barron, J. T., Mildenhall, B., Verbin, D., Srinivasan, P. P., Hedman, P., 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5470–5479.

Basso, A., Condorelli, F., Giordano, A., Morena, S., Perticarini, M., 2024. Evolution of Rendering Based on Radiance Fields. The Palermo Case Study for a Comparison Between Nerf and Gaussian Splatting. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 57–64.

Besl, P. J., McKay, N. D., 1992. Method for registration of 3-d shapes. *Sensor fusion IV: control paradigms and data structures*, 1611, Spie, 586–606.

Dubayah, R. O., Drake, J. B., 2000. Lidar remote sensing for forestry. *Journal of forestry*, 98(6), 44–46.

Goli, L., Reading, C., Sellán, S., Jacobson, A., Tagliasacchi, A., 2024. Bayes’ rays: Uncertainty quantification for neural radiance fields. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20061–20070.

Haitz, D., Hermann, M., Roth, A. S., Weinmann, M., Weinmann, M., 2024. The Potential of Neural Radiance Fields and 3D Gaussian Splatting for 3D Reconstruction from Aerial Imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 97–104.

Jäger, M., Kapler, T., Feßenbecker, M., Birkelbach, F., Hillemann, M., Jutzi, B., 2024. HoloGS: Instant Depth-based 3D Gaussian Splatting with Microsoft HoloLens 2. *arXiv preprint arXiv:2405.02005*.

Jäger, M., Landgraf, S., Jutzi, B., 2023. Density Uncertainty Quantification with NeRF-Ensembles: Impact of Data and Scene Constraints. *arXiv preprint arXiv:2312.14664*.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4), 139–1.

Kheradmand, S., Rebain, D., Sharma, G., Sun, W., Tseng, J., Isack, H., Kar, A., Tagliasacchi, A., Yi, K. M., 2024. 3D Gaussian Splatting as Markov Chain Monte Carlo. *arXiv preprint arXiv:2404.09591*.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91–110.

Martin-Brualla, R., Radwan, N., Sajjadi, M. S., Barron, J. T., Dosovitskiy, A., Duckworth, D., 2021. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7210–7219.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., Ng, R., 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1), 99–106.

Müller, T., Evans, A., Schied, C., Keller, A., 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4), 1–15.

Oechsle, M., Peng, S., Geiger, A., 2021. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5589–5599.

Petrovska, I., Jäger, M., Haitz, D., Jutzi, B., 2023. Geometric Accuracy Analysis between Neural Radiance Fields (NeRFs) and Terrestrial laser scanning (TLS). *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 153–159.

Petrovska, I., Jutzi, B., 2024. Vision through Obstacles-3D Geometric Reconstruction and Evaluation of Neural Radiance Fields (NeRFs). *Remote Sensing*, 16(7), 1188.

Petrovska, I., Jutzi, B., 2025. Seeing beyond vegetation: A comparative occlusion analysis between Multi-View Stereo, Neural Radiance Fields and Gaussian Splatting for 3D reconstruction. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 100089.

Schonberger, J. L., Frahm, J.-M., 2016. Structure-from-motion revisited. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4104–4113.

Schönberger, J. L., Zheng, E., Frahm, J.-M., Pollefeys, M., 2016. Pixelwise view selection for unstructured multi-view stereo. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, Springer, 501–518.

Tancik, M., Weber, E., Ng, E., Li, R., Yi, B., Wang, T., Kristoffersen, A., Austin, J., Salahi, K., Ahuja, A. et al., 2023. Nerfstudio: A modular framework for neural radiance field development. *ACM SIGGRAPH 2023 Conference Proceedings*, 1–12.

Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J. T., Srinivasan, P. P., 2022. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 5481–5490.

Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V. A., 2021. NeRF–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*.

Ye, V., Li, R., Kerr, J., Turkulainen, M., Yi, B., Pan, Z., Seiskari, O., Ye, J., Hu, J., Tancik, M. et al., 2024. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*.