ISPRS Open Journal of Photogrammetry and Remote Sensing 17 (2025) 100100

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing

FeatureGS: Eigenvalue-feature optimization in 3D Gaussian Splatting for geometrically accurate and artifact-reduced reconstruction

Miriam Jäger *, Markus Hillemann, Boris Jutzi

Institute of Photogrammetry and Remote Sensing (IPF), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

ARTICLE INFO

Keywords:
3D Gaussian Splatting
Point cloud
3D reconstruction
Artifact removal
Features
Eigenvalues

ABSTRACT

3D Gaussian Splatting (3DGS) has emerged as a powerful approach for 3D scene reconstruction using 3D Gaussians. However, neither the centers nor surfaces of the Gaussians are accurately aligned to the object surface, complicating their direct use in point cloud and mesh reconstruction. Additionally, 3DGS typically produces floater artifacts, increasing the number of Gaussians and storage requirements. To address these issues, we present FeatureGS, which incorporates an additional geometric loss term based on an eigenvalue-derived 3D shape feature into the optimization process of 3DGS. The goal is to improve geometric accuracy and enhance properties of planar surfaces with reduced structural entropy in local 3D neighborhoods, typically given in man-made environments. We present four alternative formulations for the geometric loss term based on 'planarity' of Gaussians, as well as 'planarity', 'omnivariance', and 'eigenentropy' of Gaussian neighborhoods. On the small-scale DTU benchmark with man-made scenes, FeatureGS achieves a 20% improvement in geometric accuracy, suppresses floater artifacts by 90%, and reduces the number of Gaussians by 95%. FeatureGS proves to be a strong method for geometrically accurate, artifact-reduced and memory-efficient 3D scene reconstruction, enabling the direct use of Gaussian centers for geometric representation.

1. Introduction

The creation of geometric 3D scene reconstructions has developed rapidly since the introduction of Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021). In NeRFs, a network implicitly describes the scene by estimating color and volume density for each position and direction. In contrast, 3D Gaussian Splatting (3DGS) offers new possibilities for 3D scene and point cloud reconstruction as it represents the scene through 3D Gaussians. These are ellipsoid-like structures, characterized by scaling, rotation, and color. During the optimization process, the 3D Gaussians are projected onto the image. To minimize the photometric error between the rendered images and the training images, the Gaussians are refined and adapted. Unlike NeRFs, Gaussians in 3DGS explicitly represent the scene where geometric information is allegedly present. Nevertheless, the centers and surfaces of Gaussians do not directly represent the object surface, which makes their direct use for 3D point cloud and mesh reconstruction impractical. In addition, the 3DGS often leads to floater artifacts, which further increase the already high number of Gaussians and thus storage requirements.

In this work, we present FeatureGS, which incorporates four different formulations of an additional geometric loss term based on eigenvalue-derived 3D shape features into the optimization process of 3DGS. 3D shape features are widely used for tasks for semantic interpretation and point cloud classification (Weinmann et al., 2015a, 2017). Thereby the 3D covariance matrix (3D structure tensor (Jutzi and Gross, 2009)), derived from a point and its local neighborhood, is well-known to characterize such shape properties (Weinmann et al., 2015a). The three eigenvalues, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, correspond to an orthogonal system of eigenvectors $(\epsilon_1, \epsilon_2, \epsilon_3)$, which indicate the direction (*rotation*) of the three ellipsoid principal axes and correspond to the extent (*scales*) of the 3D ellipsoid along the principal axes. Based on the behavior of the eigenvalues λ_1, λ_2, and λ_3 structures can be described.

FeatureGS aims to improve geometric accuracy of Gaussians and enhance properties of planar surfaces with a reduced structural entropy in local 3D neighborhoods of Gaussians, to improve geometric consistency and suppress noise in line with traditional filtering strategies for point clouds (Günen and Beşdok, 2021). Firstly, like previous flattening approaches (Guédon and Lepetit, 2024; Dai et al., 2024; Chen et al., 2024a; Huang et al., 2024), FeatureGS aims to flatten 3D Gaussians by enhancing the 'planarity' of Gaussians as 3D feature, in order to achieve higher geometric accuracy of Gaussian centers. Secondly, real physical circumstances of point clouds can be described by interpretable geometric features with a single value (Hillemann et al., 2019). To enhance the structural representation of the 3D Gaussian centers

* Corresponding author.
E-mail address: miriam.jaeger@kit.edu (M. Jäger).

https://doi.org/10.1016/j.ophoto.2025.100100
Received 6 February 2025; Received in revised form 9 August 2025; Accepted 26 August 2025
Available online 1 September 2025

in a neighborhood, particularly for man-made objects aligning with Manhattan-World-Assumption (Coughlan and Yuille, 1999, 2000), we leverage neighborhood-based 3D features derived from the k-nearest neighbors (kNN) of each Gaussian. By incorporating the 3D features either 'planarity', 'omnivariance', or 'eigenentropy' in the geometric loss, the characterization of local 3D structures with a predominance of planar surfaces with a structural entropy is reinforced.

In this contribution, we investigate whether the integration of different geometric loss terms of FeatureGS can enhance the 3D geometric accuracy of Gaussian centers and suppress floater artifacts by reinforcing specific 3D shape properties of Gaussians and Gaussian neighborhoods. The evaluation focuses on the Chamfer cloud-to-cloud distance for geometrically 3D accuracy and artifact-reduction, and the total number of Gaussians required to represent the scene for memory efficiency. While our primary goal is to achieve precise geometric representation and efficient memory usage, we additionally report the rendering quality, measured by Peak Signal-to-Noise Ratio (PSNR), to ensure consistency in scene reconstruction. Experiments are conducted on 15 scenes from the DTU benchmark dataset.

We demonstrate that FeatureGS strikes a remarkable balance between geometric accuracy, floater artifact suppression, and memory efficiency by integrating 3D shape feature properties into the optimization process of 3D Gaussian Splatting. It improves geometric accuracy by enforcing surface-aligned Gaussian distributions, enabling the Gaussian centers to serve as a more precise geometric representation. In parallel, it drastically reduces the total number of Gaussians required to represent a scene at comparable rendering quality, resulting in artifact-reduced and memory-efficient reconstructions. All variants of FeatureGS additionally benefit from faster training. Overall, FeatureGS yields the following key outcomes:

- Up to 30% improvement in geometric accuracy on the DTU benchmark with man-made objects, measured via Chamfer distance.
- Enhanced surface alignment of Gaussians, enabling direct use of their centers as geometric representation.
- Suppression of floater artifacts by up to 90% on DTU and over 55% on Tanks and Temples benchmark.
- Reduction of required Gaussians by up to 96%, leading to smaller model sizes and up to 24× compression, along with 20%–30% shorter training times compared to 3DGS.

In the following, we first discuss related work in Section 2, providing an overview of 3D feature extraction and novel view synthesis and 3D reconstruction techniques, as well as the use of Gaussian splats for 3D reconstruction. In Section 3, we present our methodology, FeatureGS, which integrates four alternative novel geometric loss formulations based on 3D shape features to enhance 3DGS. In Section 5 we introduce the experimental setup, including datasets, evaluation metrics, implementation details, and configurations for the photometric-geometric loss. In Sections 5 and 6, we present the results of our experiments on small-scale and large-scale benchmark datasets. While comparing FeatureGS to 3DGS and 2DGS with a focus on both quantitative and qualitative aspects. The derived results are discussed in Section 8, highlighting the advantages of FeatureGS concerning geometric accuracy, artifact removal and memory efficiency. Additionally, Section 7 presents an ablation study that investigates the following aspects: hyperparameter tuning of the photometric-geometric loss term, and the combination into a multi-feature loss. Finally, in Section 9, the effectiveness of FeatureGS is outlined, and future directions for optimizations and practical applications are suggested.

2. Related work

In this section, we provide an overview of the different types of 3D features in Section 2.1, which are essential for FeatureGS. Subsequently, in Section 2.2, we present a brief overview of novel view synthesis and 3D reconstructions, followed by an introduction to 3D reconstructions with Gaussian splats.

2.1. 3D features

Several types of 3D features exist for point cloud-based applications such as classification, registration, or calibration. Complex features, which cannot be interpreted directly include descriptors such as Shape Context 3D (SC3D) (Frome et al., 2004), Signature of Histogram of OrienTations (SHOT) (Tombari et al., 2010) or Fast Point Feature Histograms (FPFH) (Rusu et al., 2009). In contrast, interpretable features (Weinmann et al., 2015c) are those that are directly interpretable, such as local 2D and 3D shape features. To describe the local structure around a 3D point, the spatial arrangement of other 3D points in the local neighborhood is often considered. Thereby the 3D covariance matrix, also known as the 3D structure tensor, is well-known and suitable for characterizing the shape properties of 3D data (Weinmann et al., 2015a). It is derived explicitly for each point from the point itself and its local neighbors. The three eigenvalues, $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$, correspond to an orthogonal system of eigenvectors $(\epsilon_1, \epsilon_2, \epsilon_3)$, which indicate the direction (*rotation*) of the three ellipsoid principal axes and correspond to the extent (*scales*) of the 3D ellipsoid along the principal axes. Based on the behavior of the eigenvalues λ_1, λ_2, and λ_3, linear ($\lambda_1 \gg \lambda_2, \lambda_3$), planar ($\lambda_1 \approx \lambda_2 \gg \lambda_3$), and spherical ($\lambda_1 \approx \lambda_2 \approx \lambda_3$) structures can be described. The use of geometric 3D shape features has led to thousands of publications in various fields over the past few decades. They are especially used for the automatic semantic interpretation and classification (Weinmann et al., 2015a, 2017, 2020) of point clouds. But also for calibration (Hillemann et al., 2019) or registration (Bueno et al., 2018) of 3D point clouds.

2.2. 3D reconstruction with Gaussian Splats

The pioneering research on Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021) builds the fundament for 3D reconstructions with Gaussian Splats, and builds upon Scene Representation Networks (Sitzmann et al., 2019), which represent the scene as a function of 3D coordinates within the scene. NeRFs extend this concept by estimating color values and densities for each 3D coordinate through 6D camera positions and associating 2D images through the training of a multilayer perceptron (MLP). NeRF was followed by thousands of publications driving research and development of neural surface reconstructions, point cloud and mesh reconstruction (Oechsle et al., 2021; Wang et al., 2021; Yariv et al., 2021; Li et al., 2023; Jäger and Jutzi, 2023) in various domains. However, NeRF describes the scene implicitly by estimating a color and volume density for each position and direction, which are also subject to a certain degree of uncertainty (Jäger et al., 2025).

In contrast 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) offers new possibilities for 3D scene reconstruction. With 3DGS a novel concept of 3D scene representation was elaborated, in which a scene is explicitly represented by a large set of 3D Gaussians. Each Gaussian is defined by its mean, covariance, opacity, and spherical harmonics for color definition. The covariance is parameterized using scaling and rotation. These 3D Gaussians are projected into 2D Gaussians to the 2D image space, allowing high-quality real-time rendering. To optimize the scene, the Gaussians are initialized from a point cloud produced by Structure from Motion (SfM). The Gaussians' parameters (means for the Gaussian centers, scaling, rotations, opacities, and color) are then refined during optimization to match the training images. More Gaussians are added as needed to improve the scene representation. This optimization process leads to scenes with millions of small Gaussians that represent the 3D object geometry. Nevertheless, Gaussians do not take an ordered structure in general (Guédon and Lepetit, 2024), and the center or surface of a Gaussian does not directly align with the actual object surface. In addition, 3DGS often leads to floater artifacts, which further increase the high number of Gaussians and thus the storage requirements.

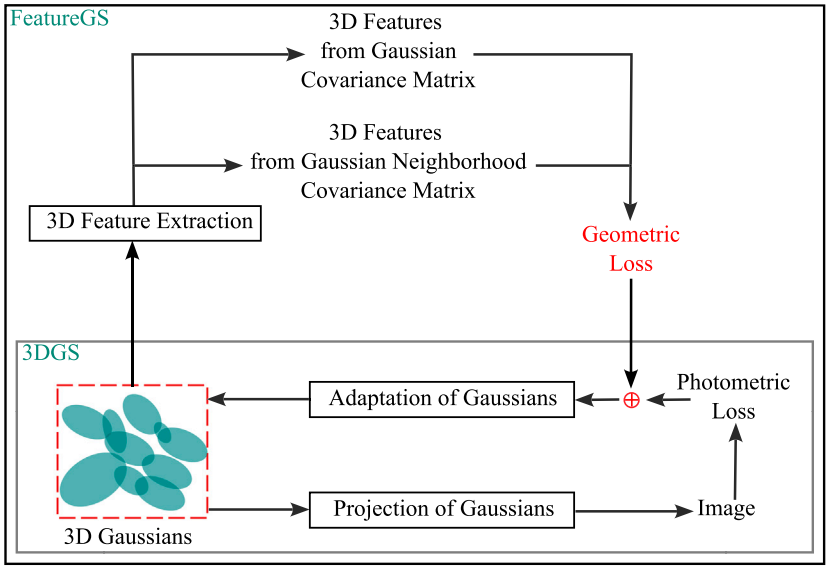


Fig. 1. Methodology of FeatureGS: Geometric loss based on 3D shape features added to 3DGS (Kerbl et al., 2023). The features are derived from eigenvalues of the covariance matrix from individual Gaussians or the covariance matrix from Gaussians in a local neighborhood surrounding each Gaussian center. The geometric loss is combined with the photometric loss in 3DGS.

The concept of transforming 3D Gaussians into 2D ellipses or planar ellipse-like structures in order to achieve higher geometric accuracy is widely used in many approaches. SuGaR (Guédon and Lepetit, 2024) extracts meshes from 3DGS by introducing a regularization term that aligns Gaussians with the scene surface. Surfels (Dai et al., 2024) combines 3D Gaussian points' optimization flexibility with the surface alignment of surfels by flattening 3D Gaussians into 2D ellipses, setting the z-scale to zero. PGSR (Chen et al., 2024a) flattens Gaussians into planes, using unbiased depth rendering to obtain precise depth information. 2DGS (Huang et al., 2024) follows a similar approach and collapses 3D volumes directly into 2D planar Gaussian disks for view-consistent geometry, using perspective-accurate splatting with ray-splat intersection and depth and normal consistency terms. MVG-Splatting (Li et al., 2024) improves 2DGS by optimizing normal calculation and using an adaptive densification method guided by depth maps. MIP-Splatting (Yu et al., 2024) introduces a 3D smoothing filter to constrain Gaussian sizes based on the input views' sampling frequency, eliminating high-frequency artifacts.

In contrast to other works, FeatureGS utilizes geometric 3D shape features to enhance specific Gaussian and Gaussian neighborhood properties for geometrically accurate and artifact-reduced 3D reconstruction. The 3D features are embedded within the optimization process of 3DGS through an additional geometric loss term in four alternative formulations into a photometric-geometric loss term. On the one hand, like previous approaches (Guédon and Lepetit, 2024; Dai et al., 2024; Chen et al., 2024a; Huang et al., 2024), FeatureGS flattens 3D Gaussians. However, FeatureGS incorporates the 3D feature 'planarity' for that. On the other hand, FeatureGS can enhance properties of planar surfaces with reduced structural entropy by utilizing 3D features 'planarity', 'omnivariance' and 'eigenentropy' in Gaussian neighborhoods.

3. Methodology

In this section, we describe FeatureGS (Fig. 1) with four alternative additional geometric loss formulations based on 3D shape features. These features are derived from the eigenvalues of the covariance matrix of the Gaussians and Gaussian neighborhoods, and provide insights into the spatial structure within both individual Gaussians and the Gaussians in a local neighborhood surrounding each Gaussian center. Our proposed geometric loss is combined with the photometric loss used in 3DGS to create a comprehensive photometric-geometric loss function.

3.1. Photometric loss

The photometric loss term of 3DGS measures the similarity between rendered images and ground truth images using pixel-level comparison metrics. This loss includes both L1 loss and a Structural Similarity Index (SSIM) term to capture differences in luminance, contrast, and structure between the images. The photometric loss is given by the following loss function (Eq. (1)).

$$L_{\text{photometric}} = (1-\theta)L_1 + \theta L_{\text{D-SSIM}} \tag{1}$$

with θ, L_1-Norm of the per pixel color difference and $L_{\text{D-SSIM}}$-Term (Kerbl et al., 2023).

3.2. Geometric loss

We introduce four different novel additional geometric loss terms, based on the eigenvalue-derived 3D shape features to enhance specific properties (Fig. 2) of 3D Gaussian itself and Gaussian centers in a neighborhood. For the first approach, we aim to flatten Gaussians to achieve a high geometric accuracy of the Gaussian centers. This is done by incorporating the 3D shape feature 'planarity' from eigenvalues (scales) (Fig. 3(a)) of each Gaussian itself in the geometric loss term. For the second approach, we incorporate a neighborhood-based geometric loss term using the 3D shape features from covariance matrix (Fig. 3(b)) by the k-nearest neighbors (kNN) of each Gaussian center. To enhance a specific characterization of local 3D structures of man-made objects aligning with Manhattan-World-Assumption (Coughlan and Yuille, 1999, 2000), we strengthen the predominance of planar surfaces, and a structural entropy. This is done through the Gaussian neighborhood 3D shape features 'planarity', 'omnivariance', and 'eigenentropy'.

3.2.1. Covariance matrix

Gaussian. 3DGS uses an explicit representation of the scene through 3D Gaussians. These ellipsoid-like structures are characterized by scaling, rotation, and color, including opacity. The scaling components can be interpreted analogously to the three eigenvalues $s_1 \geq s_2 \geq s_3 \geq 0$ and the rotation components to the eigenvectors $(\epsilon_1, \epsilon_2, \epsilon_3)$ of the covariance matrix. By using the normalized eigenvalues (scales) of the Gaussian covariance matrix (Fig. 3(a)), we compute the 3D shape feature.

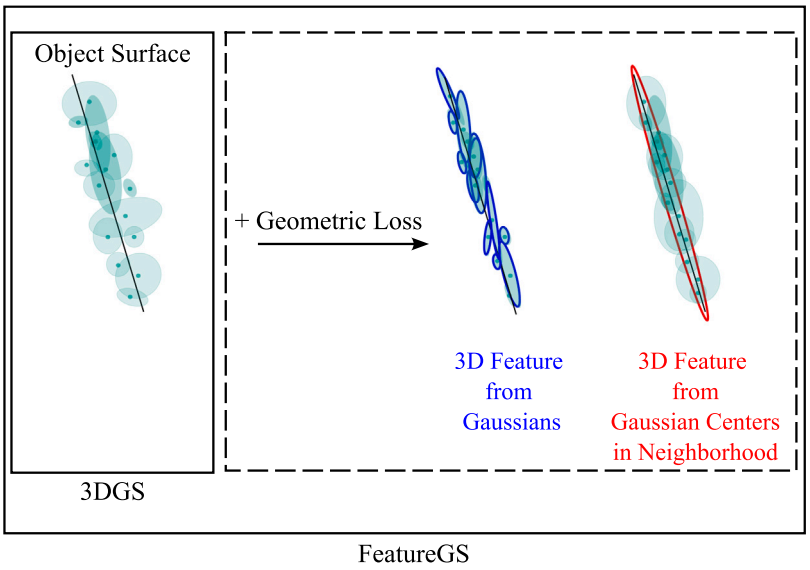


Fig. 2. Additional geometric loss of FeatureGS, illustrated by a 3D feature from Gaussians and a 3D feature from Gaussian centers in a local neighborhood. For example, through the loss with planarity of Gaussians, the Gaussians become more planar, and through the loss with planarity, omnivariance, or eigenentropy in the Gaussian neighborhood, the alignment of the Gaussian centers in the neighborhood becomes more planar. All configurations have the effect that the Gaussians move closer to the object surface and are less randomly oriented. This enables the Gaussian centers to serve as a geometric representation of the surface.

Gaussian neighborhood. Given a point p_0 in the 3D space, i.e., the center of a Gaussian, we define its k-nearest neighbors $\{p_1, p_2, \dots, p_k\}$. The centroid $\bar{p}$ (Eq. (2)) of this neighborhood is computed as:

$$\bar{p} = \frac{1}{k+1} \sum_{i=0}^{k} p_i \tag{2}$$

The covariance matrix C (Eq. (3)) (Weinmann et al., 2015) for the neighborhood (Fig. 3(b)) is then:

$$C = \frac{1}{k+1} \sum_{i=0}^{k} (p_i - \bar{p})(p_i - \bar{p})^T \tag{3}$$

From C, eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are derived, providing shape properties for the neighborhood.

3.2.2. Eigenvalue normalization

To ensure consistency, eigenvalues s_1, s_2, s_3 from the Gaussian covariance matrix and $\lambda_1, \lambda_2, \lambda_3$ from the Gaussian neighborhood covariance matrix are normalized by dividing by the sum of the eigenvalues for each case.

For the Gaussian covariance matrix (Eq. (4)):

$$s'_i = \frac{s_i}{\text{sum}(\mathbf{s})} \quad \text{for } i \in \{1,2,3\}, \tag{4}$$

with

$$\text{sum}(\mathbf{s}) = \sum_{i=1}^{3} s_i. \tag{5}$$

For the Gaussian neighborhood covariance matrix (Eq. (6)):

$$\lambda'_i = \frac{\lambda_i}{\text{sum}(\lambda)} \quad \text{for } i \in \{1,2,3\}, \tag{6}$$

with

$$\text{sum}(\lambda) = \sum_{i=1}^{3} \lambda_i. \tag{7}$$

The normalized eigenvalues s'_1, s'_2, s'_3 and $\lambda'_1, \lambda'_2, \lambda'_3$ are then ordered in descending order:

$$s'_1 \geq s'_2 \geq s'_3 \geq 0 \quad \text{and} \quad \lambda'_1 \geq \lambda'_2 \geq \lambda'_3 \geq 0.$$

The normalized eigenvalues are then used for the final geometric 3D feature computation.

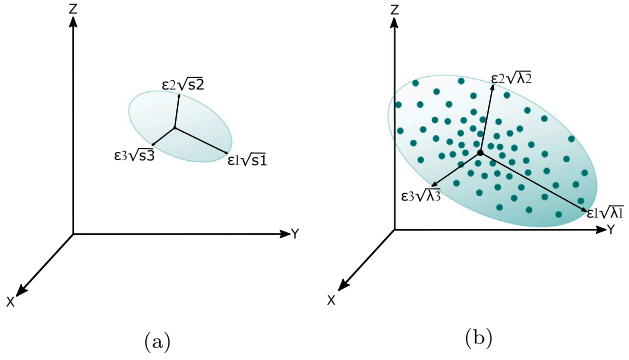


Fig. 3. a Representation of a single Gaussian ellipsoid with the three eigenvectors $(\epsilon_1, \epsilon_2, \epsilon_3)$ and the corresponding eigenvalues (s_1, s_2, s_3) in the three-dimensional coordinate system. b Representation of an ellipsoid from the neighborhood points represented by the Gaussian centers with the three eigenvectors $(\epsilon_1, \epsilon_2, \epsilon_3)$ and the corresponding eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ in the three-dimensional coordinate system.

3.2.3. Geometric loss with Gaussians

Gaussian planarity. Planarity (Weinmann et al., 2015) measures the extent to which a Gaussian resembles a planar structure. It is defined as:

$$\text{Planarity}_{\text{Gauss}} = \frac{s'_2 - s'_3}{s'_1} \tag{8}$$

The Gaussian planarity loss (Eq. (9)), preferring high planarity similar to other flattening approaches, is:

$$L_{\text{Plan,Gauss}} = \left(1 - \frac{s'_2 - s'_3}{s'_1}\right) \tag{9}$$

3.2.4. Geometric loss with Gaussian neighborhoods

To enhance the structural properties that 3D point clouds of man-made objects exhibit, we incorporate a neighborhood-based geometric loss using the k-nearest neighbors (kNN) of each point. This approach allows for the calculation of spatial features in the local neighborhood

of each Gaussian. The strengthening of the characterization of planar surfaces with reduced structural entropy in local 3D neighborhoods is achieved by including a geometric neighborhood loss. For this purpose, we consider the 3D features $\text{Planarity}_{\text{kNN}}$, $\text{Omnivariance}_{\text{kNN}}$ and $\text{Eigenentropy}_{\text{kNN}}$ in Gaussian neighborhoods of kNN from the normalized eigenvalues $\lambda'_1 \geq \lambda'_2 \geq \lambda'_3 \geq 0$, explained in more detail below.

Neighborhood planarity. Similar to the purpose of maintaining planarity of each Gaussian itself, we want to strengthen the properties of man-made objects according to the Manhattan-World-Assumption (Coughlan and Yuille, 1999, 2000) and other (almost) planar surfaces, and suppress the spherical spread of the Gaussians in a neighborhood. Therefore, in addition to the planarity of the Gaussians, we use the planarity (Weinmann et al., 2015) in the neighborhood. This is defined as:

$$\text{Planarity}_{\text{kNN}} = \frac{\lambda'_2 - \lambda'_3}{\lambda'_1} \tag{10}$$

The neighborhood planarity loss (Eq. (11)) is:

$$L_{\text{Plan,kNN}} = \left(1 - \frac{\lambda'_2 - \lambda'_3}{\lambda'_1}\right) \tag{11}$$

Neighborhood omnivariance. The omnivariance indicates the volume of the neighborhood and expresses whether the respective points scatter locally in all directions. In previous work (Weinmann et al., 2015b), the omnivariance is indicated to be a highly relevant feature of point cloud classification. Omnivariance (Weinmann et al., 2015) and the neighborhood omnivariance loss (Eq. (12)) is defined as:

$$L_{\text{Omni,kNN}} = \text{Omnivariance}_{\text{kNN}} = \sqrt[3]{\lambda'_1 \lambda'_2 \lambda'_3} \tag{12}$$

Minimizing the neighborhood omnivariance loss reduces the local scattering of the points.

Neighborhood eigenentropy. The eigenentropy quantifies the order/disorder of the local structure of the neighborhood points by measuring the entropy within the local 3D neighborhood based on the normalized eigenvalues. Additionally, it has shown to be an appropriate 3D feature for characterizing plane point cloud structures (Dittrich et al., 2017). Eigenentropy (Weinmann et al., 2015) and the neighborhood eigenentropy loss (Eq. (13)) is defined as:

$$L_{\text{Eigen,kNN}} = \text{Eigenentropy}_{\text{kNN}} = -\sum_{i=1}^{3} \lambda'_i \log(\lambda'_i) \tag{13}$$

Minimizing the neighborhood eigenentropy loss favors a minimum disorder (Weinmann et al., 2017) and therefore low entropy of 3D points.

3.3. Combined photometric-geometric loss

Our four different final loss functions L combine the conventional photometric loss $L_{\text{photometric}}$ of 3DGS with each one of four different geometric loss $L_{\text{geometric}}$ terms. This incorporates both the 3D shape properties of each Gaussian itself $L_{\text{geometric,Gaussian}}$ or the neighborhood features $L_{\text{geometric,kNN}}$ based on Gaussian centers. The photometric loss ensures the quality of pixel rendering by adjusting the Gaussians according to their projection onto the image plane, while the geometric loss term enhances specific properties of 3D structures. The total photometric-geometric loss L (Eq. (14)) is defined as:

$$L = h_{\text{photo}} \cdot L_{\text{photometric}} + L_{\text{geometric}}, \tag{14}$$

with

$$L_{\text{geometric}} \in \{L_{\text{Plan,Gaussian}}, L_{\text{Plan,kNN}}, L_{\text{Omni,kNN}}, L_{\text{Eigen,kNN}}\}. \tag{15}$$

and the hyperparameter h_{photo} for balancing the weighting between the photometric and geometric components.

4. Experiments

In this section, we present the experimental setup. We introduce the used datasets (Section 4.1), the evaluation metrics (Section 4.2), the setted implementation details (Section 4.3), as well as the configurations for our photometric-geometric loss (Section 4.4).

4.1. Dataset

Small-scale dataset. For the evaluation of FeatureGS, we use the small-scale DTU benchmark dataset (Jensen et al., 2014). The dataset consists of scenes featuring real objects, including either 49 or 64 RGB images, corresponding camera poses, and reference point clouds obtained from a structured-light scanner (SLS). We specifically focus on the same 12 scenes as previous approaches (Dai et al., 2024; Chen et al., 2024a; Huang et al., 2024; Li et al., 2024).

Large-scale dataset. Additionally, we evaluate on the Tanks and Temples dataset (Knapitsch et al., 2017), which contains large-scale outdoor and indoor scenes with complex geometry and varied lighting conditions in real-world environments. Reference point clouds are given, obtained from an industrial laser scanner. We specifically focus on the four outdoor scenes barn, caterpillar, ignatius and truck. All experiments were performed using the automatic image resolution downscaling applied by 3DGS in its default configuration, in order to match the memory constraints of the used GPU.

4.2. Metrics

To evaluate our method quantitatively and qualitatively, we report the 3D geometric accuracy, the number of Gaussians needed to represent the scene for memory efficiency and the rendering quality. For 3D evaluation we report the Chamfer cloud-to-cloud distance. To evaluate surface accuracy, we use the DTU evaluation procedure (Jensen et al., 2014), which masks out points above 10 mm. In addition, we use the Chamfer cloud-to-cloud distance for all points to evaluate the presence of floater artifacts external to the object. Low Chamfer distance indicates high accuracy and less artifacts. Gaussian storage requirements by the total numbers of Gaussian. We evaluate the 2D rendering quality of the images with the Peak Signal-to-Noise Ratio (PSNR) in dB, whereby a high PSNR is targeted. For the large-scale Tanks and Temples dataset, training time and model size are additionally tracked to evaluate the capability of FeatureGS for challenging, real-world environments.

4.3. Implementation details

3D Gaussian Splatting[1] is processed according to the original implementation, using default densification strategies and the default parameters with learning rates of 0.0025 for spherical harmonics features, 0.05 for opacity adjustments, 0.005 for scaling operations and 0.001 for rotation transformations, on a NVIDIA RTX3090 GPU.

2D Gaussian Splatting[2] is processed according to the original implementation by using default parameters.

Firstly, we consider the same number of training iterations of 15 000, which are recommended from 3DGS. For a fair comparison, we consider the evaluation procedure by training with early stopping on each the same reached PSNR value. This should demonstrate that FeatureGS enables pushing down the total numbers of Gaussians representing the scene, while achieving higher geometric accuracy and artifact-reduced rendering for the same photometric rendering quality.

[1] https://github.com/graphdeco-inria/gaussian-splatting (last access 07/21/2024).

[2] https://github.com/hbb1/2d-gaussian-splatting (last access 04/29/2025).

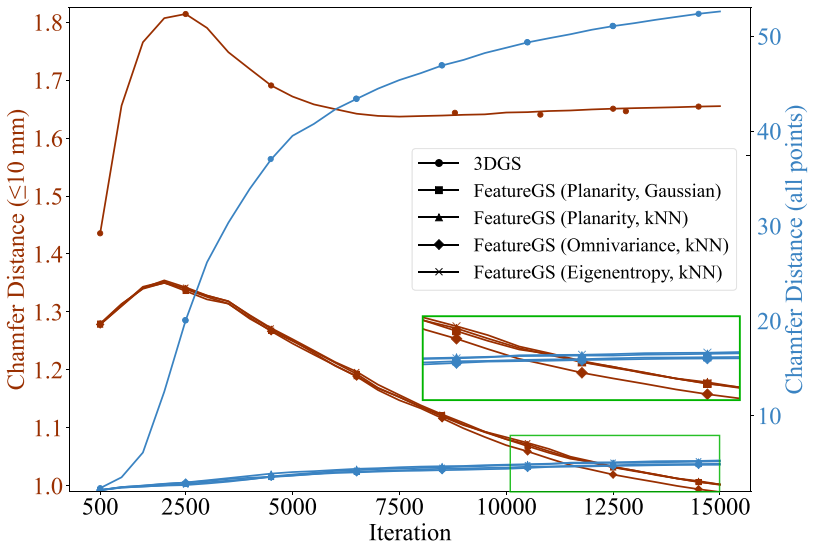


Fig. 4. Geometric accuracy during training process on the DTU scan40 for different loss types. Chamfer cloud-to-cloud distances ↓ in mm for points ≤10 mm and all points (floater artifacts).

4.4. Loss configurations

The photometric loss for the optimization is given by the loss function in Eq. (1) with $\theta = 0.2$ by default (Kerbl et al., 2023).

For the final different photometric-geometric loss formulations (Eq. (14)) of FeatureGS the weighting with $h_{\text{photo}} = 0.05$ is chosen. This is based on hyperparameter tuning (Fig. 12) in Section 7.1, to create a proper balance between rendering quality and geometric accuracy, approximately where the reached PSNR remains the same but the Chamfer cloud-to-cloud distance increases.

As the 3D distribution of the Gaussians and hence their centers changes through the optimization, we decide on a fixed number of kNN = 50 (Weinmann et al., 2017) as representative k-nearest neighbors. Through the variable distribution and density of the points during the training process, we aim to achieve an effect similar to multi-scale (Brodu and Lague, 2012) neighborhoods, which have proven to be robust in point cloud classification tasks.

5. Results: Small-scale data

The following sections show qualitative (Section 5.1) and quantitative (Section 5.2) results of FeatureGS in comparison to 3DGS and 2DGS. We distinguish between the training with a fixed number of training iterations in Section 5.1.2 and a fixed achievable rendering quality in Section 5.1.3, represented by an early stopping of the PSNR. This should demonstrate the performance of FeatureGS in terms of geometric accuracy, floater artifact-reduction, memory efficiency, and yet strong rendering quality, based on the two criteria.

5.1. Quantitative results

5.1.1. Training process

Over the training process of 15 000 iterations, the original photometric loss of 3DGS and the photometric-geometric loss of FeatureGS demonstrate distinct behaviors in terms of geometric accuracy, presence of floater artifacts, number of Gaussians representing the scene, and rendering quality.

It is observed that the Chamfer cloud-to-cloud distance (Fig. 4) for 3DGS continuously increases for all points during training process. For instance, in the case of scene40, the distance rises to approximately 50 mm. In contrast, for all geometric FeatureGS losses, the distance remains consistently low throughout the training process. Only a slight increase is present, which is due to the fact that the initial point cloud from SfM nonetheless has the highest accuracy and FeatureGS also reconstructs points that are not in the (incomplete) reference point cloud. For scan40, this distance stabilizes at around 4–5 mm. This indicates that, unlike FeatureGS, the 3DGS training process incorporates a significant number of points (see Fig. 5) with higher geometric inaccuracies. Regarding the geometric surface accuracy, measured by masking out points with errors over 10 mm, the distance for 3DGS initially increases to approximately 1.9 mm, then decreases and stabilizes at a constant value. For scan40, the distance starts at approximately 1.2 mm, peaks at 1.9 mm, and eventually stabilizes at 1.7 mm. Conversely, the distance for FeatureGS increases less at the start of training and then decreases further as training progresses. For scan40, it decreases from approximately 1.3 mm to 1.0 mm.

With regard to rendering quality (Fig. 6), as measured by PSNR (and SSIM), the original photometric loss of 3DGS significantly outperforms the combined photometric-geometric loss of FeatureGS. For 3DGS, the PSNR continuously increases and appears to converge after approximately 14 000 training iterations. In contrast, for all FeatureGS loss functions, the PSNR initially increases rapidly but saturates at a noticeably lower value after about 10 000 iterations. The behavior of SSIM follows a similar trend.

5.1.2. Fixed training iterations

The following quantitative results for the fixed number of training iterations of 15 000 provide the geometric accuracy by Chamfer cloud-to-cloud distance, the number of resulting Gaussians, and the rendering quality reported by the PSNR.

Geometric accuracy. For the geometric accuracy of the surface points (Table 1), which are located at a distance of 10 mm from the reference point cloud $L_{\text{Plan,Gauss}}$ and $L_{\text{Eigen,kNN}}$ yield often the best and second best highest geometric accuracies. $L_{\text{Plan,kNN}}$ and $L_{\text{Omni,kNN}}$ achieve a mixed result, but show good performance in some scenes such as scan24, scan37. Nevertheless, the differences between all geometric-radiometric FeatureGS configurations are mostly marginal and stable across all scans (see e.g. scan55 with Chamfer distances from 0.967 to 0.971 mm). This is also reflected in the mean geometric accuracy. Compared to 2DGS (1.331 mm) and 3DGS (1.609 mm), FeatureGS achieves the highest geometric accuracy with a mean Chamfer distance between 1.310 and 1.315 mm.

Floater artifacts due to presumably incorrectly reconstructed Gaussians external to the actual object, where smaller values mean less disturbing artifacts, are illustrated by Table 2. Regarding the reduction

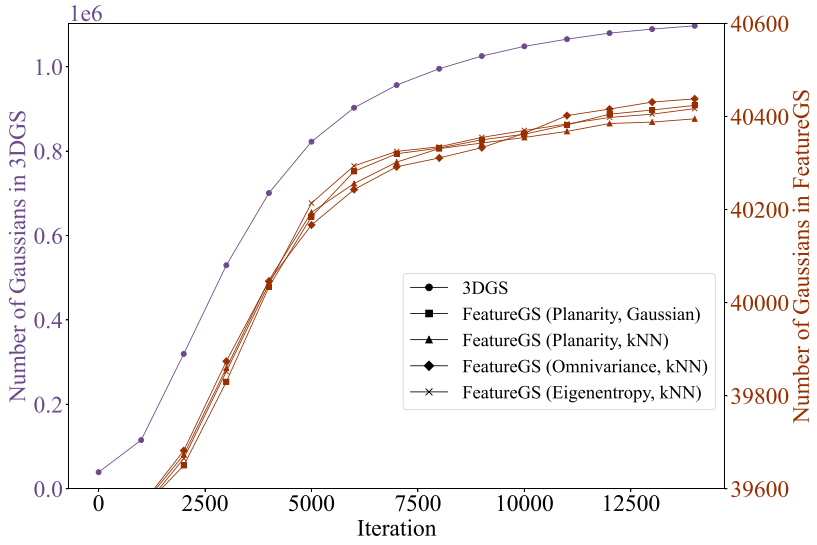


Fig. 5. Numbers of Gaussians during training process on the DTU scan40 for different loss types.

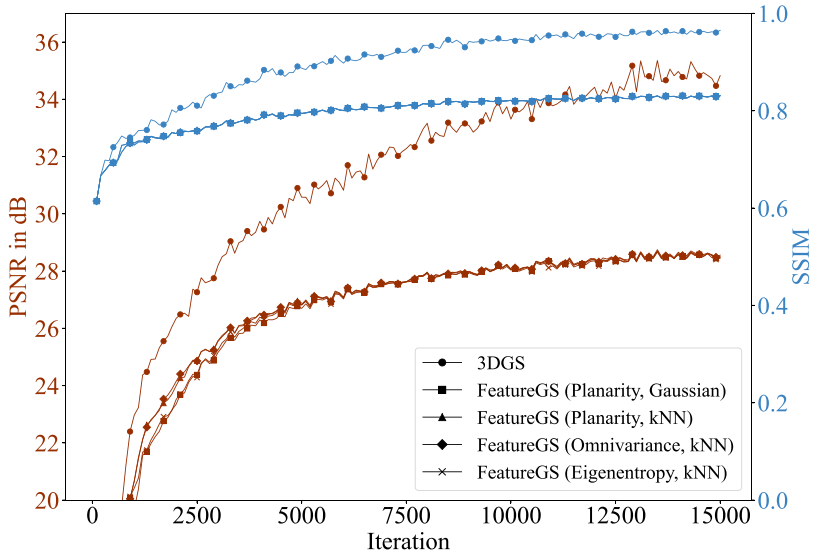


Fig. 6. Rendering quality during training process on the DTU scan40 for different loss types. Peak Signal-to-Noise Ratio (PSNR) ↑ in dB and SSIM ↑.

of floater artifacts, $L_{\text{Plan, Gaussian}}$ and $L_{\text{Eigen,kNN}}$ prove to be particularly effective. $L_{\text{Plan,Gauss}}$ often achieves the best results and shows a strong ability to minimize floater artifacts, especially for scans such as scan40 (4.816 mm) and scan55 (4.782 mm). On average, $L_{\text{Plan,Gauss}}$ performs best with 10.593 mm, followed by $L_{\text{Plan,kNN}}$ with 10.793 mm. Overall, there is a significant improvement in all FeatureGS configurations compared to 3DGS, both in terms of surface accuracy and floater reduction. FeatureGS reduces the mean Chamfer distance for surface accuracy by around 0.3 mm (approx. 20% improvement). In particular, compared to 3DGS (116.587 mm) and 2DGS (95.359 mm), FeatureGS reduces floater artifacts by over 90%.

Number of Gaussians. Table 3 shows the number of Gaussians generated by 3DGS and the different loss configurations of FeatureGS. The mean values indicate that all the FeatureGS configurations reduce the number of Gaussians by around 440 000 Gaussians on average, which corresponds to a reduction of around 95%. 2DGS, while more efficient than 3DGS, produces nearly eight times more Gaussians than FeatureGS. FeatureGS increases the number of initial points by only around 7%. The relative differences between the FeatureGS configurations are only minor. All FeatureGS configurations deliver a consistently clear reduction compared to 3DGS.

Rendering quality. While FeatureGS is significantly more memory efficient (fewer Gaussians, less storage required), has less floater artifacts and delivers geometrically more accurate results, there are drawbacks in rendering quality (Table 4). On average, the mean PSNR values appear lower with a decrease in rendering quality of approximately 3.3 dB. The differences between the different FeatureGS loss formulations are minimal (less than 0.1 dB).

5.1.3. Fixed rendering quality

The quantitative results for the fixed PSNR using early stopping demonstrate the geometric accuracy due to the Chamfer cloud-to-cloud distance and the number of Gaussians required for this. The comparison of 3DGS and the FeatureGS configurations with identical PSNR serves to evaluate different aspects of the methods under comparable rendering qualities. This ensures that differences in other metrics such as geometric accuracy, number of Gaussians or floater artifacts are not influenced by a varying of the rendering quality.

Geometric accuracy. FeatureGS consistently outperforms 3DGS in geometric accuracy of surface points (Table 5) for the same rendering quality, with an average improvement of about 30%, with a mean

Table 1
Surface accuracy. **Geometric accuracy** comparison on the DTU dataset with Chamfer cloud-to-cloud distances ↓ in mm for points ≤10 mm from the reference, according to the DTU evaluation script. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Methods	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
scan24	1.702	1.229	1.421	1.438	1.434	1.432
scan37	1.782	1.525	1.324	1.309	1.317	1.360
scan40	1.625	0.955	1.002	1.002	0.989	1.001
scan55	1.361	0.746	0.969	0.968	0.967	0.971
scan63	2.061	2.023	1.483	1.449	1.462	1.481
scan65	1.708	1.441	1.518	1.526	1.506	1.513
scan69	1.671	1.276	1.299	1.316	1.312	1.314
scan83	2.285	2.024	1.428	1.425	1.417	1.412
scan97	1.855	1.759	1.689	1.684	1.695	1.689
scan105	1.778	1.508	1.172	1.168	1.163	1.180
scan106	1.514	0.758	0.936	0.939	0.948	0.950
scan110	1.486	1.356	1.819	1.821	1.800	1.808
scan114	1.549	0.941	0.966	0.952	0.960	0.945
scan118	1.291	0.681	0.875	0.854	0.873	0.866
scan122	1.289	0.743	0.992	1.000	0.991	0.990
Mean	1.609	1.331	**1.313**	**1.310**	**1.311**	1.315

Table 2
Floater artifacts. **Geometric accuracy** comparison on the DTU dataset with Chamfer cloud-to-cloud distances↓ in mm are reported for all points to focus on floaters external to the point cloud. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Methods	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
scan24	50.850	45.800	8.909	9.378	21.158	12.469
scan37	53.919	65.792	9.575	8.312	9.435	9.045
scan40	43.597	65.314	4.915	5.174	4.816	5.267
scan55	58.004	69.473	5.050	5.990	4.782	5.059
scan63	279.172	172.740	19.130	24.350	20.405	22.034
scan65	179.180	170.143	17.916	15.357	19.246	18.741
scan69	121.251	112.102	10.110	9.653	9.708	9.770
scan83	178.645	141.350	24.628	21.874	22.426	21.545
scan97	111.836	65.862	13.099	12.033	11.333	9.755
scan105	132.986	68.334	8.221	8.159	8.260	8.480
scan106	88.501	80.452	3.272	3.459	3.058	3.211
scan110	164.030	120.396	17.160	14.134	17.584	18.517
scan114	173.681	135.634	5.850	6.773	6.138	6.002
scan118	83.070	85.573	6.977	6.374	7.005	7.087
scan122	124.686	101.461	9.332	9.755	9.265	9.369
Mean	116.587	95.359	**10.593**	**10.793**	12.212	**11.721**

Table 3
Number of Gaussians on the DTU dataset. We report the total number of Gaussians ↓ resulting from the four alternative loss formulations of FeatureGS, compared to 3DGS, 2DGS, and the number of SfM points used for initialization. Mean scores are listed at the bottom. Best results (lowest total number) concerning memory are highlighted as 1st, 2nd, and 3rd. The training incorporates 15 000 iterations.

Methods	3DGS	2DGS	FeatureGS				Initial SfM
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$	
scan24	673 276	318 556	20 105	20 423	20 485	20 440	15 479
scan37	766 722	390 183	29 431	29 111	29 247	29 291	24 857
scan40	831 896	351 751	40 425	40 404	40 445	40 429	39 158
scan55	739 171	327 276	34 760	34 774	34 738	34 780	33 506
scan63	249 496	145 388	13 343	13 461	13 610	13 509	10 869
scan65	347 906	158 023	14 231	14 154	14 213	14 216	13 203
scan69	304 854	164 466	15 931	15 911	15 906	15 911	15 264
scan83	216 765	119 774	11 982	11 921	12 054	11 913	10 652
scan97	595 899	252 064	22 699	22 717	22 579	22 436	20 467
scan105	250 257	131 301	26 102	26 154	26 111	26 210	25 291
scan106	269 773	114 440	33 701	33 707	33 696	33 705	33 523
scan110	227 484	93 552	11 822	11 768	11 835	11 838	11 382
scan114	361 373	140 436	26 208	26 248	26 226	26 199	25 761
scan118	357 583	156 268	27 964	27 948	27 973	27 967	27 650
scan122	318 226	145 922	21 427	21 423	21 417	21 417	20 975
Mean	462 699	189 427	**24 275**	**24 277**	24 302	**24 291**	22 771

Table 4
Rendering quality comparison on the DTU dataset. We report the PSNR ↑ in dB. Mean scores are listed at the bottom. Best results are highlighted as 1st, 2nd, and 3rd. The training incorporates 15 000 iterations.

Methods	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
scan24	35.16	32.03	29.86	29.90	29.98	29.93
scan37	29.98	28.51	26.32	26.35	26.39	26.36
scan40	34.59	31.76	28.48	28.52	28.56	28.45
scan55	34.08	31.87	29.48	29.55	29.50	29.56
scan63	37.35	35.43	32.65	32.66	32.74	32.81
scan65	35.19	31.78	30.35	30.35	30.36	30.35
scan69	33.50	30.91	28.33	28.57	28.54	28.53
scan83	34.08	31.12	32.69	32.82	32.84	31.85
scan97	32.57	30.99	30.03	30.06	30.12	30.01
scan105	36.70	34.41	34.68	34.64	34.51	34.59
scan106	37.48	36.17	36.01	36.09	36.06	36.03
scan110	31.81	30.19	29.94	29.96	29.92	29.94
scan114	34.78	32.55	32.73	32.55	32.70	32.64
scan118	36.71	35.08	34.81	34.85	34.82	34.83
scan122	36.06	34.83	34.15	34.17	34.15	34.17
Mean	**34.67**	**32.51**	31.37	31.40	**31.41**	31.34

Table 5
Surface accuracy. **Geometric accuracy** comparison on the DTU dataset with Chamfer cloud-to-cloud distances ↓ in mm for surface points ≤10 mm from the reference, according to the DTU evaluation script. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates iterations until early stopping at same PSNR.

Methods	3DGS	FeatureGS			
		$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
scan24	2.026	1.424	1.463	1.475	1.446
scan37	1.847	1.297	1.313	1.278	1.280
scan40	1.758	0.954	0.948	0.952	0.951
scan55	1.672	0.935	0.918	0.944	0.914
scan63	2.155	1.530	1.534	1.504	1.500
scan65	2.095	1.589	1.576	1.581	1.582
scan69	1.916	1.288	1.271	1.290	1.271
scan83	2.211	1.438	1.489	1.507	1.509
scan97	1.912	1.680	1.699	1.704	1.704
scan105	1.769	1.264	1.332	1.293	1.280
scan106	1.574	1.095	1.104	1.104	1.100
scan110	1.902	1.866	1.854	1.831	1.853
scan114	1.453	1.010	1.015	1.008	1.022
scan118	1.503	1.064	1.063	1.053	1.063
scan122	1.604	1.060	1.070	1.060	1.051
Mean	1.826	**1.300**	1.310	**1.306**	**1.302**

Table 6
Floater artifacts. **Geometric accuracy** comparison on the DTU dataset with Chamfer cloud-to-cloud distances ↓ in mm are reported for all points to focus on floaters external to the point cloud. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates iterations until early stopping at same PSNR.

Methods	3DGS	FeatureGS			
		$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
scan24	32.241	11.835	8.441	14.137	9.151
scan37	72.622	9.451	11.153	8.361	8.852
scan40	19.356	4.796	5.475	5.191	5.751
scan55	36.010	5.233	4.872	4.727	5.199
scan63	200.478	20.862	19.600	20.744	22.942
scan65	163.601	17.775	18.103	15.096	17.191
scan69	61.014	9.524	9.5613	9.736	9.561
scan83	139.395	22.819	22.671	21.552	23.454
scan97	70.390	11.543	12.408	12.085	11.960
scan105	80.220	8.102	8.037	8.067	8.172
scan106	32.873	3.228	3.021	3.031	3.206
scan110	111.052	16.205	15.742	16.463	16.771
scan114	57.211	6.006	5.732	6.078	5.409
scan118	58.760	5.199	5.172	4.870	4.814
scan122	270.132	8.983	9.504	9.155	8.688
Mean	93.690	10.771	**10.633**	**10.620**	**10.741**

geometric accuracy of 1.826 mm for 3DGS to 1.300 to 1.310 mm for FeatureGS. The different loss formulations of FeatureGS show only minimally different results with differences of less than 1 percent.

FeatureGS heavily reduces floater artifacts (Table 6) at the same rendering quality by an average of 90% compared to 3DGS. This is shown in the mean Chamfer Distance, which is reduced from 93.690 mm for 3DGS to 10.620–10.771 mm for FeatureGS. Overall, the photometric-geometric loss formulations with $L_{\text{Plan,kNN}}$ and $L_{\text{Omni,kNN}}$ result in the lowest amount of floater artifacts.

Number of Gaussians. In addition, the number of Gaussians (Table 7) is reduced by FeatureGS while maintaining the same rendering quality compared to 3DGS. FeatureGS drastically reduces the number of Gaussians by around 90% from an average of 249 986 Gaussians to between 26 380 and 26 389 Gaussians. This leads to a lower memory requirement. Within FeatureGS, the variants show equivalent compression of the number of Gaussians.

5.2. Qualitative results

Similar to the quantitative results, FeatureGS yields promising results in terms of geometric accuracy of the 3D point clouds as well as rendering quality by removing floater artifacts. Through all geometric loss terms of FeatureGS, consistently accurate and floater artifact-reduced results are generated across all 15 scenes, compared to 3DGS. All results are shown for the exact same PSNR values by early stopping, thus the same rendering quality.

Geometric accuracy. The geometric accuracy of Gaussian centers on DTU dataset, evaluated using the Chamfer cloud-to-cloud distance (Fig. 7), highlights the superior performance of FeatureGS compared to 3DGS. For the FeatureGS, the configurations that yielded the highest surface accuracy for the respective scenes are visualized. It is important to note that the reference point clouds are incomplete, which leads to high values on all object edges. On the one hand, it shows that the accuracy of the surface points is higher for FeatureGS, achieving submillimeter accuracy. Furthermore, the surface points appear less noisy. On the other hand, the drastic reduction in floater artifacts is striking, whereby floater artifacts prevent the reconstruction of the geometry via direct extraction of the Gaussian centers. While 3DGS leads to a lot of floater artifacts in all scenes, the scenes with FeatureGS are almost artifact-free.

FeatureGS prioritizes surface accuracy and reduces floater artifacts by enhancing properties of planar surfaces with low structural entropy

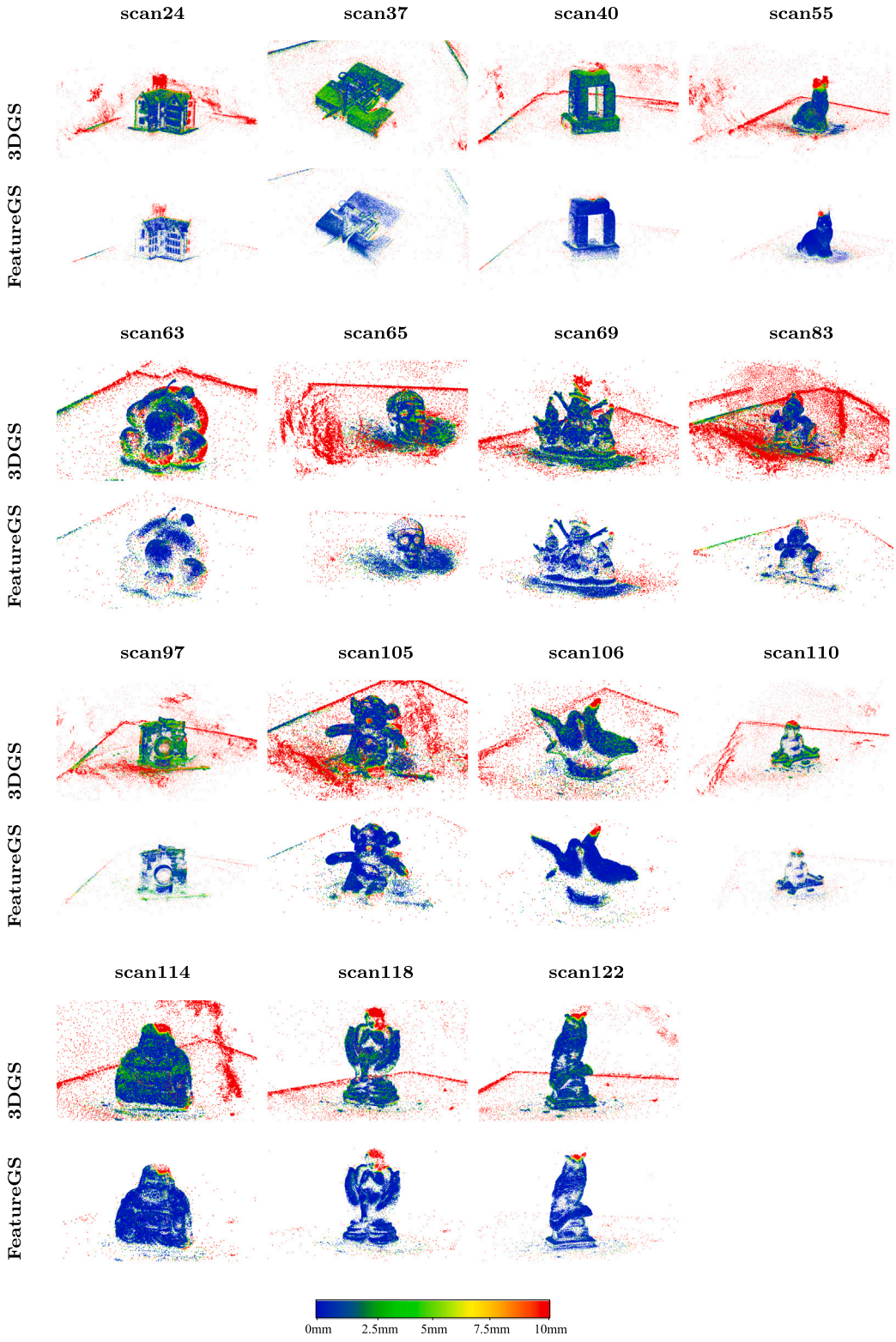
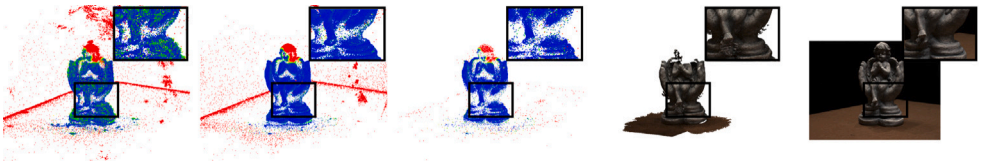


Fig. 7. Geometric accuracy comparison on the DTU dataset with Chamfer cloud-to-cloud distances ↓ for the same PSNR. Color values are cropped at 10 mm distance.

instead of increasing point density. As a consequence, the resulting point clouds remain relatively sparse. In low-textured regions, similar to the behavior observed in 3DGS and 2DGS, a smaller number of larger Gaussians is sufficient to achieve accurate photometric representation, as demonstrated in DTU scenes 24, 37, 63, 97, 110, and 118 (Fig. 7). This leads to the appearance of gaps in homogeneous areas, as shown in Fig. 8.

Rendering quality. The rendering quality (Fig. 9) shown by the rendered test images also underlines the overall strong performance of



(a) 3DGS (b) 2DGS (c) FeatureGS (d) reference (e) GT image

Fig. 8. Illustration of gaps in the point clouds from Gaussian centers at low-textured, homogeneous areas using 3DGS, 2DGS and FeatureGS. Compared to the reference point cloud and GT image for DTU scene scan118.

Table 7
Number of Gaussians on the DTU dataset. We report the total number of Gaussians ↓ resulting from the four alternative loss formulations of FeatureGS, compared to 3DGS and the number of SfM points used for initialization. Mean scores are listed at the bottom. Best results (lowest total number) concerning memory are highlighted as 1st, 2nd, and 3rd. The training incorporates iterations until early stopping at same PSNR.

Methods	3DGS	FeatureGS				Initial SfM
		$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eig,kNN}}$	
scan24	333 870	20 491	20 100	20 553	20 155	15 479
scan37	527 713	29 199	29 596	29 371	28 972	24 857
scan40	537 082	40 353	40 406	40 446	40 466	39 158
scan55	470 449	34 788	34 744	34 742	34 769	33 506
scan63	113 493	13 346	13 323	13 183	13 155	10 869
scan65	151 776	14 213	14 176	14 187	14 179	13 203
scan69	147 690	15 893	15 908	15 905	15 908	15 264
scan83	132 898	11 926	12 044	11 892	12 046	10 652
scan97	303 676	22 454	22 824	22 732	22 712	20 467
scan105	184 679	26 166	26 158	26 149	26 178	25 291
scan106	125 419	33 730	33 719	33 724	33 731	33 523
scan110	140 378	11 819	11 802	11 811	11 824	11 382
scan114	190 674	26 247	26 232	26 255	26 228	25 761
scan118	165 017	27 927	27 926	27 910	27 902	27 650
scan122	172 589	21 427	21 432	21 436	21 407	20 975
Mean	249 986	26 389	**26 385**	**26 380**	**26 387**	22 771

Table 8
Surface accuracy. **Geometric accuracy** comparison on the Tanks and Temples dataset with Chamfer cloud-to-cloud distances ↓ in m for points ≤0.1 m from the reference, according to the DTU evaluation script. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Scene	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
barn	0.0276	0.0226	0.0256	0.0256	0.0257	0.0256
caterpillar	0.0161	0.0135	0.0231	0.0233	0.0235	0.0232
ignatius	0.0129	0.0104	0.0139	0.0138	0.0139	0.0139
truck	0.0137	0.0106	0.0168	0.0169	0.0168	0.0168
Mean	**0.070**	**0.014**	**0.079**	0.080	0.080	0.080

FeatureGS compared to 3DGS. The results on the FeatureGS configurations that yielded the highest floater reduction for the respective scenes are shown. It is evident that the geometric loss terms of FeatureGS significantly reduce the floater artifacts while maintaining the same quantitative rendering quality. Large dark floater artifacts disappear in hardly all scenes. In addition, the scenes appear smoother, which can be seen, e.g., in the subsoil of objects. Since the PSNR values are the same, the high PSNR value is supposedly due to the focus being on rendering the object itself and not overfitting the entire scene, which causes the creation of floater artifacts. It can also be seen that the floaters that were visible in the figures of the geometric accuracies (Fig. 7) are actually also clearly present in the synthetically rendered results. Therefore, they cannot only be removed by filtering the Gaussians with e.g. very small opacity values. In addition, FeatureGS also removes artifacts which merge with the object surface and leads to a kind of smoothing effect, such as in scan55 or scan69.

6. Results: Large-scale data

The following sections show qualitative (Section 6.1) and quantitative (Section 6.2) results of FeatureGS in comparison to 3DGS and 2DGS.

6.1. Quantitative results

The results across the key metrics, geometric accuracy, floater reduction, number of Gaussians, model size, rendering quality, as well as training time, are presented in Tables 9–13.

A clear advantage of FeatureGS lies in the substantial suppression of floater artifacts (Table 9). While 3DGS and 2DGS exhibit high average floater distances of 16.17 m and 17.61 m respectively, all FeatureGS formulations reduce floater artifacts by over 55%. The distances for FeatureGS range from 7.28 m to 7.32 m across variants, with the best result achieved by $L_{\text{Plan,kNN}}$, while even the least effective FeatureGS variant performs better than 3DGS and 2DGS.

FeatureGS leads to a drastic reduction in the number of Gaussians (Table 10). While 3DGS and 2DGS require on average 2.05 million and 1.26 million Gaussians respectively, FeatureGS reduces this to as few as 84,880, depending on the loss variant. This corresponds to a reduction of 96% compared to 3DGS and 93% compared to 2DGS, leading to a more memory-efficient model representation. This reduction is directly reflected in the model size (Table 11). While 3DGS and 2DGS models average around 485 MB and 304 MB respectively, FeatureGS produces compact models of 20.00 MB, depending on the loss variant. This corresponds to a compression factor of up to 24× compared to 3DGS and 15× compared to 2DGS, resulting in a substantially more compact representation.

Despite this reduction, FeatureGS maintains competitive rendering quality on the large-scale data (Table 12). The mean PSNR of all FeatureGS variants ranges from 23.50 dB to 23.67 dB, closely matching 2DGS (23.00 dB) and staying within 1 dB of 3DGS (24.56 dB). In terms of surface accuracy (Table 8), FeatureGS variants show higher Chamfer distances (0.079–0.080 m) compared to 2DGS (0.014 m) and 3DGS (0.070 m), which reflects a slight degradation in geometric precision for the large-scale data with diverse object types.

Nonetheless, FeatureGS demonstrates efficient training performance compared to both 3DGS and 2DGS (Table 13). While 3DGS requires on average 12.77 min and 2DGS 17.03 min, all FeatureGS loss formulations complete training in less time for the same 15 000 iterations, with mean durations between 9.12 and 10.67 min. The fastest variant, $L_{\text{Plan,Gauss}}$, benefits from directly operating on features derived from Gaussian eigenvalues without neighborhood queries. The kNN-based variants remain efficient due to accelerated implementations via PyTorch3D.[3]

[3] https://github.com/facebookresearch/pytorch3d (last accessed: 08/02/2024).

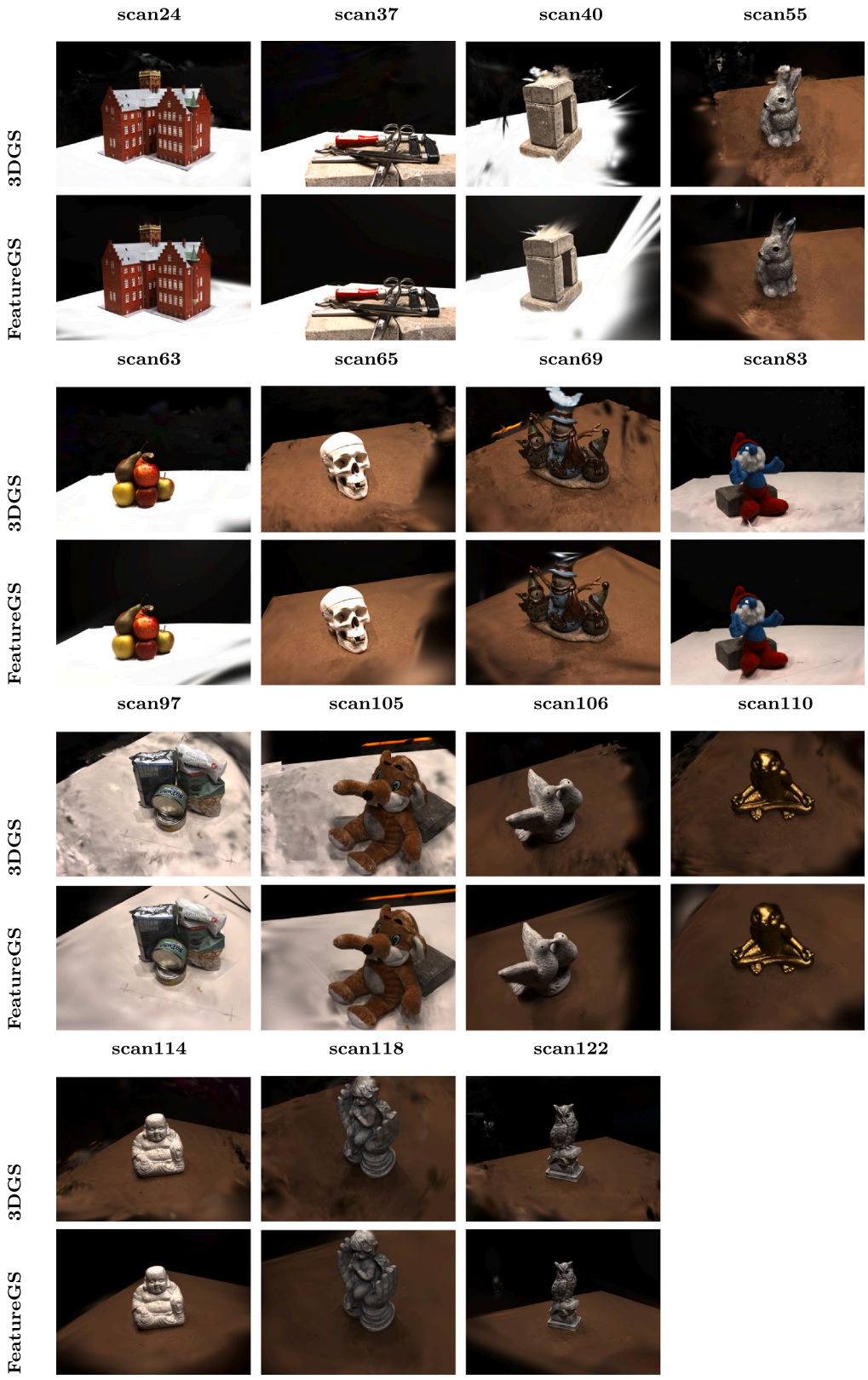


Fig. 9. Rendering quality comparison on the DTU dataset for the same PSNR.

6.2. Qualitative results

The geometric accuracy of Gaussian centers on Tanks and Temples dataset, evaluated using the Chamfer cloud-to-cloud distance (Fig. 10), show that 3DGS generates a dense set of Gaussians, where the object structure is barely discernible from the distribution of Gaussian centers. The Gaussians form a broad layer around the object surface, extending both in front of and behind it. In contrast, 2DGS produces a denser point cloud with higher accuracy and significantly fewer Gaussians, whose centers lie close to the surface.

Table 9
Floater artifacts. **Geometric accuracy** comparison on the Tanks and Temples dataset with Chamfer cloud-to-cloud distances↓ in m are reported for all points to focus on floaters external to the point cloud. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Scene	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
barn	8.44	11.04	5.33	5.33	5.32	5.32
caterpillar	20.91	22.62	8.02	8.04	8.06	8.05
ignatius	16.28	16.51	8.06	8.00	7.99	8.03
truck	19.04	18.27	7.83	7.73	7.83	7.86
Mean	16.17	17.61	**7.31**	**7.28**	**7.30**	7.32

Table 10
Number of Gaussians on the Tanks and Temples dataset. We report the total number of Gaussians ↓ of FeatureGS, compared to 3DGS, 2DGS. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Scene	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
barn	941 645	550 030	78 868	78 870	78 836	78 915
caterpillar	1 244 008	732 518	100 626	100 703	100 637	100 624
ignatius	3 255 203	2 080 397	96 258	96 155	96 103	96 199
truck	2 753 361	1 660 639	63 800	63 792	64 023	63 913
Mean	2 048 554	1 258 396	**84 888**	**84 880**	**84 900**	84 913

Table 11
Model size in MB on the Tanks and Temples dataset. We report the mean model size ↓ of FeatureGS, compared to 3DGS, 2DGS. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Scene	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
barn	222	180	18.6	18.6	18.6	18.6
caterpillar	293	173	23.7	23.8	23.8	23.7
ignatius	769	474	22.7	22.7	22.7	22.7
truck	655	388	15.0	15.0	15.1	15.1
Mean	484.75	303.75	**20.00**	**20.03**	**20.05**	**20.05**

Table 12
Rendering quality comparison on the Tanks and Temples dataset. We report the PSNR ↑ in dB. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom. The training incorporates 15 000 iterations.

Scene	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
barn	25.89	24.70	25.45	25.47	25.28	25.45
caterpillar	24.11	22.50	23.91	23.87	23.66	23.80
ignatius	22.92	20.11	21.43	21.55	21.46	21.11
truck	25.31	24.70	23.70	23.79	23.78	23.65
Mean	**24.56**	23.00	**23.62**	**23.67**	23.55	23.50

Table 13
Training time comparison on the Tanks and Temples dataset. We report the minutes for 15 000 iterations. Best results are highlighted as 1st, 2nd, and 3rd. Mean scores are listed at the bottom.

Scene	3DGS	2DGS	FeatureGS			
			$L_{\text{Plan,Gauss}}$	$L_{\text{Plan,kNN}}$	$L_{\text{Omni,kNN}}$	$L_{\text{Eigen,kNN}}$
barn	11.24	15.17	9.73	10.34	10.39	10.31
caterpillar	10.89	15.78	9.53	11.30	11.30	11.28
ignatius	14.77	19.00	8.41	10.63	10.66	10.64
truck	14.17	18.17	8.79	10.26	10.31	10.11
Mean	12.77	17.03	**9.12**	**10.63**	10.67	**10.59**

FeatureGS yields a clearly defined and precise object structure with high surface accuracy; most points lie directly on the surface, with few positioned in front of or behind the reference. The point cloud remains relatively sparse, and FeatureGS exhibits limitations in scenes with strong linear structures, such as scene caterpillar, where 2DGS better reconstructs thin structures. For most man-made or planar environments, such as facades or grounds, FeatureGS demonstrates a qualitative better geometric accuracy on the object surface. Comparing the rendered images (Fig. 11) of 3DGS and FeatureGS, 3DGS better represents backgrounds with vegetation, such as trees. Nevertheless, FeatureGS strengthen man-made planar object parts, and successfully reduces the floater artifacts. Aside from this, visual differences are minimal, as the tanks and temples datasets provide extensive image coverage, resulting in negligible floaters.

Additionally, Section 7 presents an ablation study that investigates the following aspects: hyperparameter tuning of the photometric-geometric loss term, and the combination into a multi-feature loss.

7. Ablation study

This Section 7 presents an ablation study that investigates the following aspects: hyperparameter tuning of the photometric-geometric loss term (Section 7.1) and the combination into a multi-feature loss (Section 7.2).

7.1. Geometric loss

For the final different photometric-geometric loss formulations of FeatureGS the weighting based on hyperparameter h_{photo} is necessary to create a proper balance between rendering quality and geometric accuracy.

The Chamfer cloud-to-cloud distance and PSNR for varying weights of the photometric loss h_{photo} is presented in Fig. 12. As h_{photo} increases, the Chamfer distance over all points increases (from 2.047 mm at $h_{\text{photo}} = 0.01$ to 14.993 mm at $h_{\text{photo}} = 0.10$), indicating a decrease in geometric accuracy. The Chamfer distance for points within 10 mm slightly increase from 0.968 to 1.060 mm. The PSNR shows an improvement from 26.898 to 28.681 dB. That suggest that lower values for h_{photo} improve geometric accuracy, while higher values enhance image quality at the cost of increased Chamfer distance and less accurate geometry. Therefore, the weight should be optimized according to the specific application of FeatureGS. Alternatively, achieving a higher PSNR with high geometric accuracy may require more training iterations.

7.2. Combined geometric loss

All four proposed geometric loss formulations of FeatureGS share a common goal: encouraging Gaussians to better align with the underlying surface. Although they rely on different 3D shape features, their effects are largely consistent and differ only marginally (Section 5). While the loss using 'planarity' of Gaussians improves surface accuracy and the 'omnivariance' loss of Gaussian neighborhoods reduces floater artifacts on the DTU dataset, both losses target similar geometric properties. We investigate whether combining (Eq. (16)) them can meaningfully leverage their strengths to improve overall performance, with

$$L = h_{\text{photo}} \cdot L_{\text{photometric}} + L_{\text{Plan,Gaussian}} + L_{\text{Omni,kNN}} \tag{16}$$

and the hyperparameter h_{photo}.

Table 14 reports the results on several DTU scenes. The ablation study shows that the combined loss yields complementary results across the metrics. The combined loss reduces floater artifacts and the number of Gaussians but does not further improve the surface accuracy, while the PSNR remains high and comparable to the best single-loss configurations.

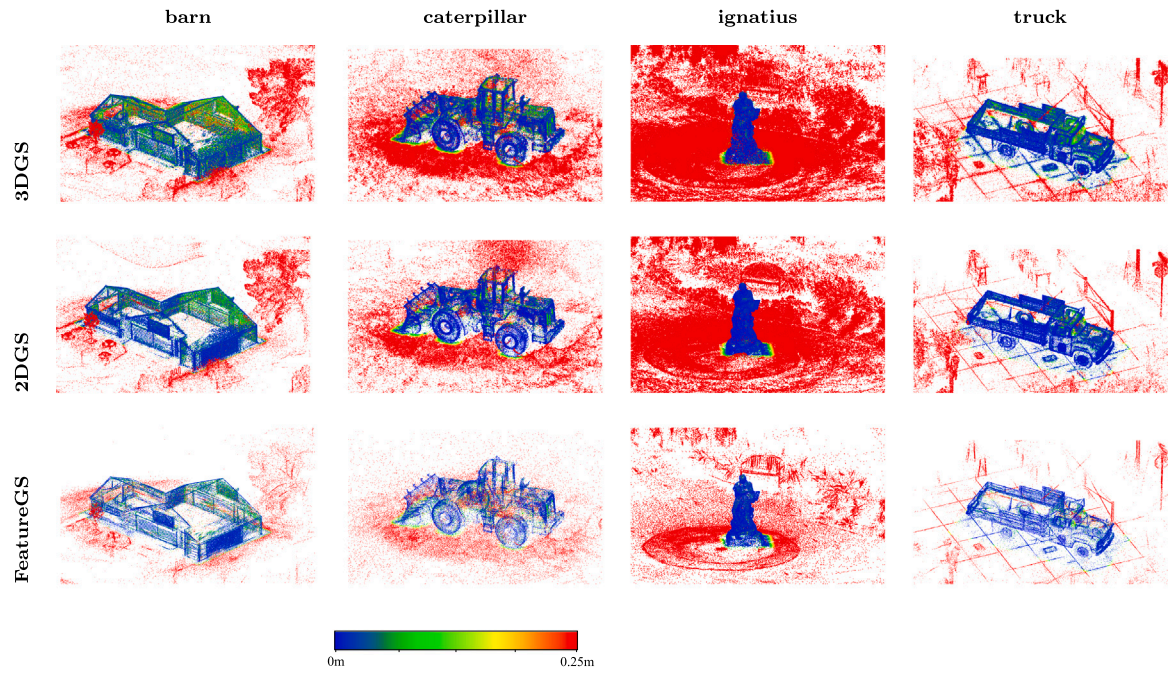


Fig. 10. Geometric accuracy comparison on the Tanks and Temples dataset with Chamfer cloud-to-cloud distances ↓. Color values are cropped at 0.25 m distance. The training incorporates 15 000 iterations.

Fig. 11. Rendering quality comparison of 3DGS and FeatureGS on the Tanks and Temples dataset, as well as ground truth (GT) images in original resolution.

8. Discussion

The evaluation of FeatureGS demonstrates its advantages in terms of geometric accuracy, reduction of floater artifacts, and memory efficiency. Although these improvements coincide with a moderate reduction in rendering quality, the results illustrate that FeatureGS is particularly suitable for diverse applications, especially those involving scenes dominated by man-made structures.

On the small-scale DTU dataset, using a fixed number of 15 000 training iterations, FeatureGS yields substantial improvements in geometric reconstruction accuracy, reducing the mean Chamfer cloud-to-cloud distance by approximately 20% compared to 3DGS. Among the tested variants, the loss formulations $L_{\text{Plan,Gauss}}$ and $L_{\text{Eigen,kNN}}$ consistently result in the highest geometric accuracy. However, differences among the four FeatureGS loss configurations remain relatively small and stable across the diverse scenes, suggesting that the

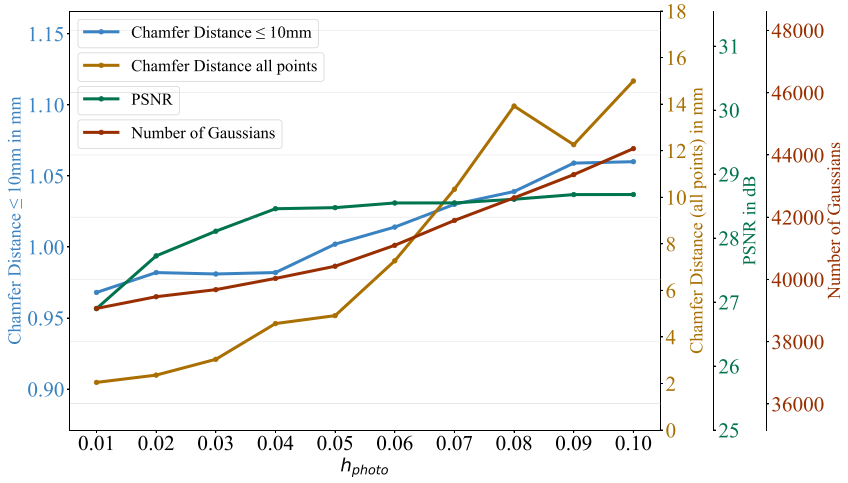


Fig. 12. Chamfer cloud-to-cloud Distances, PSNR and Number of Gaussians at different weighting configurations of the photometric-geometric loss term with varying h_{photo}.

Table 14
Multi-feature loss. Comparison on DTU FeatureGS with a combination of two alternative loss formulations $L_{\text{Plan,Gauss}}$ and $L_{\text{Omni,kNN}}$. The table shows geometric accuracy on surface accuracy and floater artifacts with Chamfer cloud-to-cloud distances↓ in mm, PSNR in dB, and number of Gaussians.

		scan105	scan106	scan110	scan114	scan118
Surface	$L_{\text{Plan,Gauss}}$	1.172	0.936	1.819	0.966	0.875
	$L_{\text{Omni,kNN}}$	1.163	0.948	1.800	0.960	0.873
	Combined	1.223	0.937	1.883	0.955	0.911
Floater	$L_{\text{Plan,Gauss}}$	8.221	3.272	17.160	5.850	6.977
	$L_{\text{Omni,kNN}}$	8.260	3.058	17.584	6.138	7.005
	Combined	4.129	3.024	9.544	4.383	3.747
PSNR	$L_{\text{Plan,Gauss}}$	34.68	36.01	29.94	32.73	34.81
	$L_{\text{Omni,kNN}}$	34.51	36.06	29.92	32.70	34.82
	Combined	32.41	35.46	29.74	32.42	34.48
Gaussians	$L_{\text{Plan,Gauss}}$	26 102	33 701	11 822	26 208	27 964
	$L_{\text{Omni,kNN}}$	26 111	33 696	11 835	26 226	27 973
	Combined	25 334	33 647	11 649	26 077	27 784

method exhibits robust performance irrespective of scene complexity considering man-made objects. In addition to improved accuracy, FeatureGS achieves a considerable reduction in floater artifacts, suppressing them by approximately 90% relative to 3DGS. The formulation with $L_{\text{Plan,Gauss}}$ shows the strongest effect in reducing these artifacts. Furthermore, FeatureGS leads to a drastic decrease in the number of Gaussians, averaging a reduction of 95%. This contributes to a clearly higher memory efficiency, enhancing the suitability of FeatureGS for large-scale reconstruction scenarios. However, these gains are accompanied by a moderate decrease in rendering quality, with an average PSNR drop of around 3.3 dB. To address the inherent trade-off between geometric accuracy and photometric rendering quality, the weighting of the loss hyperparameter can be adjusted according to the intended application. When controlling rendering quality through early stopping to match the PSNR values between the loss variants, FeatureGS continues to show improvements in geometric accuracy, reducing the mean Chamfer distance by approximately 0.5 mm (≈30%) relative to 3DGS. Floater artifact suppression remains effective at around 90% across all loss formulations, and the number of Gaussians is still reduced by approximately 90%. This demonstrates that FeatureGS preserves its geometric accuracy, floater reduction and memory advantages for the same photometric performance. The qualitative results further confirm these trends. Across all evaluated scenes, FeatureGS consistently improves the geometric surface accuracy of reconstructed point clouds, while effectively removing non-surface floaters. These artifacts are likewise absent from the rendered images. Since PSNR is held constant in this setting, we suggest that the geometric losses encourage the model to allocate representational capacity toward surface-aligned geometry, rather than overfitting the background with unstructured floater artifacts.

The evaluation on the heterogeneous large-scale Tanks and Temples benchmark reveals both the strengths and the limitations of FeatureGS. The dataset comprises a diverse range of scenes, including structured, man-made environments with planar geometries (e.g., facades, grounds), as well as unstructured regions with natural vegetation and volumetric elements. For these man-made environments, FeatureGS provides a geometrically accurate but sparse representation. Similar to 2DGS, which regularization can potentially lead to over-smoothing (Huang et al., 2024) in some regions. In structured regions, FeatureGS yields high geometric accuracy and coherent reconstructions. The employed geometric loss terms effectively promote surface-aligned and compact Gaussian representations, which are well-suited to the underlying scene characteristics. In contrast, performance degrades in vegetative areas. Facades and ground in man-made environments can be geometrically distinguished from vegetation due to their structural divergence (Weinmann et al., 2013). Since the geometric loss, which favors compact, surface-aligned Gaussians and thus emphasizes planar structures, FeatureGS struggles in scenes where the spherical arrangement of Gaussians is highly irregular or volumetric, as is typical for vegetation. Despite these challenges, FeatureGS maintains a consistent advantage in the suppression of floater artifacts across all scene types. While methods such as PGSR struggles with floating points (Chen et al., 2024a) in some scenes, which is also the case for 2DGS and especially 3DGS as the results show.

Limitations and Trade-offs. While FeatureGS demonstrates strong performance across multiple benchmarks and offers clear advantages in terms of geometric accuracy, memory efficiency, and floater artifact removal, several challenges remain. *Rendering quality:* A central limitation of FeatureGS lies in its reduced rendering quality when training budgets are constrained. On the DTU benchmark, FeatureGS achieves a maximum PSNR of 31.41 dB ($L_{\text{Omni,kNN}}$), compared to 34.67 dB for 3DGS. This performance gap of more than 3 dB highlights the cost of incorporating geometric structure. A similar drop is observed for 2DGS (32.51 dB), suggesting that additional regularization and 3D shape feature-based supervision inherently trades off rendering fidelity. However, using early stopping with fixed PSNR values, FeatureGS achieves a 30% improvement in geometric accuracy and reduces memory usage by 90% relative to 3DGS. These results highlight a fundamental trade-off between photometric quality and geometric accuracy when optimizing under fixed number of training iterations. *Scene diversity:* While FeatureGS performs robustly in man-made scenes characterized by structured geometry and planar surfaces, its performance degrades in vegetative or highly unstructured regions. In such environments, which are present in the Tanks and Temples benchmark, the assumptions underlying the geometric loss, particularly the preference for planar, surface-aligned Gaussians, become less effective. The resulting reconstructions exhibit reduced fidelity in irregular, non-man-made areas, indicating a limitation in generalizing to scenes that deviate from the structural priors embedded in the geometric loss design. *Geometric coverage:* The results show that, similar to 3DGS and also 2DGS, which densification strategy favors texture-rich over geometry-rich areas (Huang et al., 2024), FeatureGS produces visible gaps in low-textured regions (e.g., DTU scenes 24, 37, 63, 97, 110, 118; Fig. 8). This sparsity is primarily due to the use of relatively few but large Gaussians in homogeneous areas. While this suffices for accurate photometric rendering, it leads to incomplete surface coverage when using Gaussian centers as geometric representation and is further reinforced in FeatureGS by the high reduction of the required number of Gaussians. To increase spatial density, one option is to adopt the original 3DGS densification strategy, which enhances Gaussian splitting in low-texture regions typically represented by few larger Gaussians. Alternatively, mesh extraction methods such as Marching Cubes (Lorensen and Cline, 1998), as used in previous work (Guédon and Lepetit, 2024; Chen et al., 2024a; Huang et al., 2024), can provide a different geometric surface representation compared to point clouds, e.g., via TSDF fields (Curless and Levoy, 1996) derived from depth maps. *Memory efficiency:* FeatureGS achieves substantial memory efficiency across both small and large-scale datasets. On the Tanks and Temples benchmark, for instance, it reduces the average model size from 484.75 MB (3DGS) and 303.75 MB (2DGS) to as little as 20.00 MB, corresponding to a compression factor exceeding 24× relative to 3DGS. This positions FeatureGS alongside recent compact representations such as LightGaussian (Fan et al., 2024), which reports a compression to 22 MB. However, FeatureGS does not match the extreme compression achieved by HAC (Chen et al., 2024b), which reduces models to as little as 8.10 MB in selected scenes. However, FeatureGS does not primarily aim for maximal compression, but rather balances memory efficiency with geometric accuracy and robustness to floater artifacts. *Loss variant:* Across both datasets, it remains challenging to identify a universally superior loss formulation among the four FeatureGS variants. On the DTU dataset, where scenes predominantly consist of planar, man-made structures, specific insights emerge: the loss based on 'planarity' of Gaussians yields the highest geometric accuracy, 'omnivariance' of Gaussian neighborhoods most effectively suppresses floater artifacts. Nevertheless, all four variants share similar structural objectives: encouraging Gaussians to represent planar surfaces and reduce structural entropy in local 3D neighborhoods. Consequently, observed differences in reconstruction accuracy, rendering quality, training time, and memory usage remain minor, and the combination of multiple loss terms (Section 7.2) does not yield consistent gains. Addressing different scene types and their characteristics, the numerical differences remain small in the performance of the four loss formulations between the scenes, e.g., in the diverse DTU dataset. All scenes vary in material properties and texturness. In general, each geometric loss consistently moves Gaussians closer to the object surface and reduces their random orientation, an effect that appears largely independent of scene-specific properties. Consequently, no consistent pattern emerges that would allow for reliably predicting which loss function performs better for a given scene type (e.g., on reflective surfaces) or for recommending a specific application scenario for each loss function. From a training time perspective, the variant ($L_{\text{Plan,Gauss}}$) from Gaussian itself is consistently the fastest, as it avoids neighborhood computations when using Gaussian neighborhoods with $L_{\text{Plan,kNN}}$, $L_{\text{Omni,kNN}}$ and $L_{\text{Eigen,kNN}}$. Overall, all variants are substantially faster than both 3DGS and 2DGS, primarily due to the reduced number of required Gaussians. And given the marginal performance differences, all four loss configurations are considered roughly equivalent.

9. Conclusion

FeatureGS extends 3D Gaussian Splatting by integrating 3D shape feature properties into the optimization process with additional geometric loss terms. It achieves consistent improvements in geometric accuracy, floater artifact suppression, and memory efficiency. While these gains are accompanied by a moderate decrease in rendering quality, the results highlight FeatureGS as a compact and geometry-aware method for geometrically accurate and floater artifact-reduced scene reconstruction.

Geometric accuracy. For man-made environments of the DTU dataset, FeatureGS improves the geometric accuracy by reducing the mean Chamfer distance by up to 30% relative to 3DGS under matched PSNR (early stopping) and up to 20% using a fixed number of training iterations. Across all four loss configurations, 'planarity' from Gaussians performs the best, while differences between variants remain small, indicating stable performance across scenes with man-made structures. **Floater artifacts.** FeatureGS suppresses non-surface floater artifacts by around 90% on DTU and by over 55% on Tanks and Temples compared to 3DGS, both quantitatively and qualitatively. These improvements persist across all loss variants and even when rendering quality is held constant, showing that the losses effectively promote surface-aligned Gaussians. **Number of Gaussians.** FeatureGS reduces the number of Gaussians by up to 95% on DTU and by up to 96% on Tanks and Temples compared to 3DGS. This leads to smaller model sizes, down to 20 MB on Tanks and Temples dataset, a 24× reduction relative to 3DGS. The reduced number of Gaussians also leads to improved training efficiency across all FeatureGS variants, which achieve training times approximately 20%–30% lower than 3DGS. Notably, the 'planarity' from Gaussian itself avoids computationally expensive neighborhood queries and achieves the highest relative speedup by 30%.

Overall, FeatureGS achieves structurally consistent representations, with improved geometric accuracy, and reduced floater artifacts and number of required Gaussians. While performance declines in unstructured, volumetric ares, e.g., vegetative scenes, the method is particularly well-suited for efficient and geometrically accurate reconstruction of structured, planar surfaces in man-made environments.

CRediT authorship contribution statement

Miriam Jäger: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Markus Hillemann:** Writing – review & editing, Validation, Methodology, Conceptualization. **Boris Jutzi:** Writing – review & editing, Supervision, Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Brodu, N., Lague, D., 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. ISPRS J. Photogramm. Remote Sens. 68, 121–134.

Bueno, M., Bosché, F., González-Jorge, H., Martínez-Sánchez, J., Arias, P., 2018. 4-plane congruent sets for automatic registration of as-is 3d point clouds with 3d bim models. Autom. Constr. 89, 120–134.

Chen, D., Li, H., Ye, W., Wang, Y., Xie, W., Zhai, S., Wang, N., Liu, H., Bao, H., Zhang, G., 2024a. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. arXiv preprint arXiv:2406.06521.

Chen, Y., Wu, Q., Lin, W., Harandi, M., Cai, J., 2024b. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In: European Conference on Computer Vision. Springer, pp. 422–438.

Coughlan, J.M., Yuille, A.L., 1999. Manhattan world: Compass direction from a single image by bayesian inference. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2, IEEE, pp. 941–947.

Coughlan, J., Yuille, A.L., 2000. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. Adv. Neural Inf. Process. Syst. 13.

Curless, B., Levoy, M., 1996. A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. pp. 303–312.

Dai, P., Xu, J., Xie, W., Liu, X., Wang, H., Xu, W., 2024. High-quality surface reconstruction using gaussian surfels. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11.

Dittrich, A., Weinmann, M., Hinz, S., 2017. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3d point cloud data. ISPRS J. Photogramm. Remote Sens. 126, 195–208.

Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z., et al., 2024. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. Adv. Neural Inf. Process. Syst. 37, 140138–140158.

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In: Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III 8. Springer, pp. 224–237.

Guédon, A., Lepetit, V., 2024. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5354–5363.

Günen, M.A., Beşdok, E., 2021. Comparison of point cloud filtering methods with data acquired by photogrammetric method and rgb-d sensors. Int. J. Eng. Geosci. 6 (3), 125–135.

Hillemann, M., Weinmann, M., Mueller, M.S., Jutzi, B., 2019. Automatic extrinsic self-calibration of mobile mapping systems based on geometric 3d features. Remote. Sens. 11 (16), 1955.

Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S., 2024. 2D gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–11.

Jäger, M., Jutzi, B., 2023. 3D density-gradient based edge detection on neural radiance fields (nerfs) for geometric reconstruction. arXiv preprint arXiv:2309.14800.

Jäger, Miriam, Landgraf, Steven, Jutzi, Boris, 2025. Density uncertainty quantification with NeRF-Ensembles: Impact of data and scene constraints. Internat. J. Appl. Earth Observat. Geoinformat. 137, 104406.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H., 2014. Large scale multi-view stereopsis evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 406–413.

Jutzi, B., Gross, H., 2009. Nearest neighbour classification on laser point clouds to gain object structures from buildings. Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 38 (Part 1), 4–7.

Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., 2023. 3D gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. 42 (4), 1–14.

Knapitsch, A., Park, J., Zhou, Q.-Y., Koltun, V., 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Trans. Graph. 36 (4).

Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.-Y., Lin, C.-H., 2023. Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465.

Li, Z., Yao, S., Chu, Y., Garcia-Fernandez, A.F., Yue, Y., Lim, E.G., Zhu, X., 2024. Mvg-splatting: Multi-view guided gaussian splatting with adaptive quantile-based geometric consistency densification. arXiv preprint arXiv:2407.11840.

Lorensen, W.E., Cline, H.E., 1998. Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal Graphics: Pioneering Efforts that Shaped the Field. pp. 347–353.

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. Commun. ACM 65 (1), 99–106.

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

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE, pp. 3212–3217.

Sitzmann, V., Zollhöfer, M., Wetzstein, G., 2019. Scene representation networks: Continuous 3d-structure-aware neural scene representations. Adv. Neural Inf. Process. Syst. 32.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique signatures of histograms for local surface description. In: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11. Springer, pp. 356–369.

Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W., 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689.

Weinmann, M., Jäger, M.A., Wursthorn, S., Jutzi, B., Hübner, P., 2020. 3D indoor mapping with the microsoft hololens: qualitative and quantitative evaluation by means of geometric features. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. 1, 165–172.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015a. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J. Photogramm. Remote Sens. 105, 286–304.

Weinmann, M., Jutzi, B., Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3d point cloud data. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. 2, 313–318.

Weinmann, M., Jutzi, B., Mallet, C., 2017. Geometric features and their relevance for 3d point cloud classification. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. 4, 157–164.

Weinmann, M., Mallet, C., Hinz, S., Jutzi, B., 2015b. Efficient interpretation of 3d point clouds by assessing feature relevance. AVN–Allg Vermess-Nachr 10 (2015), 308–315.

Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F., Jutzi, B., 2015c. Contextual classification of point cloud data by exploiting individual 3d neigbourhoods. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. II-3/W4 2 (W4), 271–278.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015. Distinctive 2d and 3d features for automated large-scale scene analysis in urban areas. Comput. Graph. 49, 47–57.

Yariv, L., Gu, J., Kasten, Y., Lipman, Y., 2021. Volume rendering of neural implicit surfaces. Adv. Neural Inf. Process. Syst. 34, 4805–4815.

Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A., 2024. Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19447–19456.