

# PGSR: Planar-based Gaussian Splatting for Efficient and High-Fidelity Surface Reconstruction

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai,  
Nan Wang, Haomin Liu, Hujun Bao, Guofeng Zhang<sup>†</sup>

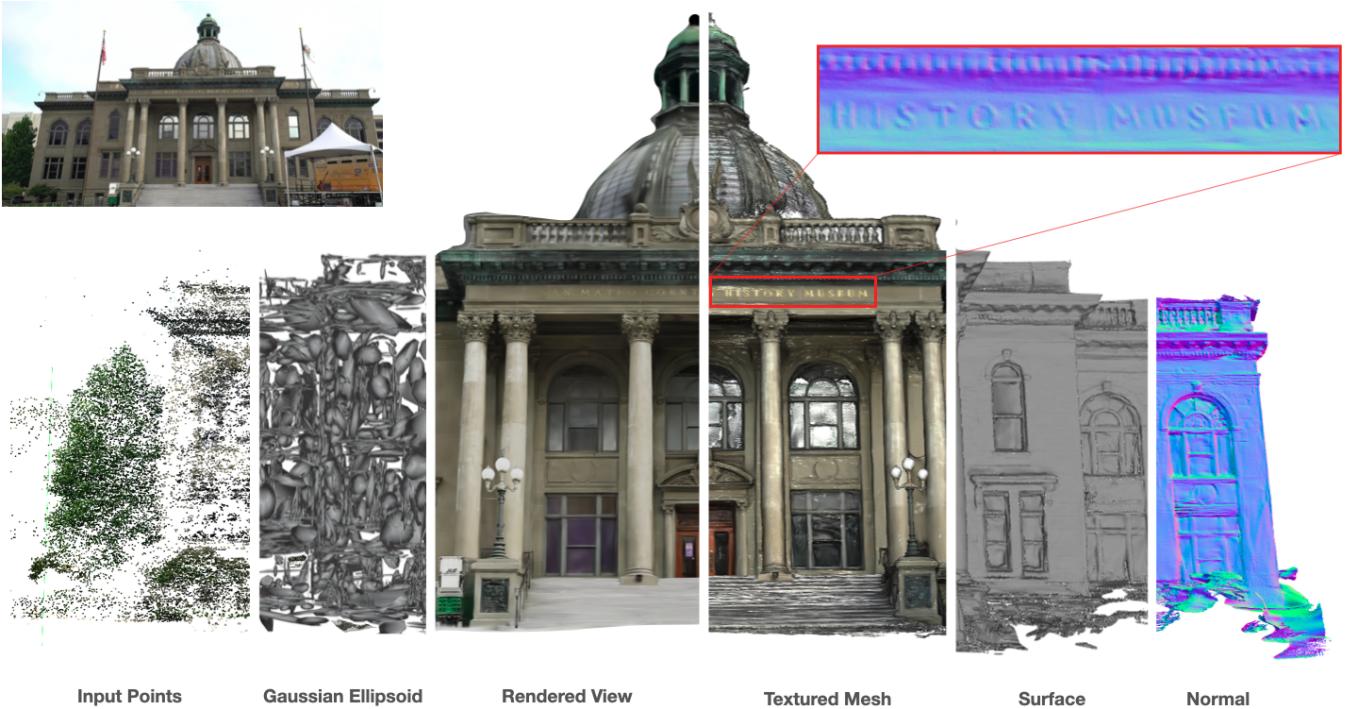


Fig. 1: **PGSR representation.** We present a Planar-based Gaussian Splatting Reconstruction representation for efficient and high-fidelity surface reconstruction from multi-view RGB images without any geometric prior (depth or normal from pre-trained model). The courthouse reconstructed by our method demonstrates that PGSR can recover geometric details, such as textual details on the building. From left to right: input SfM points, planar-based Gaussian ellipsoid, rendered view, textured mesh, surface, and normal.

**Abstract**—Recently, 3D Gaussian Splatting (3DGS) has attracted widespread attention due to its high-quality rendering, and ultra-fast training and rendering speed. However, due to the unstructured and irregular nature of Gaussian point clouds, it is difficult to guarantee geometric reconstruction accuracy and multi-view consistency simply by relying on image reconstruction loss. Although many studies on surface reconstruction based on 3DGS have emerged recently, the quality of their meshes is generally unsatisfactory. To address this problem, we propose a fast planar-based Gaussian splatting reconstruction representa-

H. Bao, G. Zhang, W. Ye and H. Li are with the State Key Lab of CAD&CG, Zhejiang University. E-mails: {baohujun, zhangguofeng}@zju.edu.cn.

D. Chen and W. Xie are with the State Key Lab of CAD&CG, Zhejiang University and SenseTime Research. D. Chen is also affiliated with Tetras.AI. E-mails: chendanpeng@tetras.ai, xiweijian@sensetime.com.

Y. Wang is with Shanghai AI Laboratory.

S. Zhai, N. Wang and H. Liu are with SenseTime Research.

<sup>†</sup> Corresponding author

tion (PGSR) to achieve high-fidelity surface reconstruction while ensuring high-quality rendering. Specifically, we first introduce an unbiased depth rendering method, which directly renders the distance from the camera origin to the Gaussian plane and the corresponding normal map based on the Gaussian distribution of the point cloud, and divides the two to obtain the unbiased depth. We then introduce single-view geometric, multi-view photometric, and geometric regularization to preserve global geometric accuracy. We also propose a camera exposure compensation model to cope with scenes with large illumination variations. Experiments on indoor and outdoor scenes show that our method achieves fast training and rendering while maintaining high-fidelity rendering and geometric reconstruction, outperforming 3DGS-based and NeRF-based methods. Our code will be made publicly available, and more information can be found on our project page (<https://zju3dv.github.io/pgsr/>).

**Index Terms**—Planar-Based Gaussian Splatting, Surface Reconstruction, Neural Rendering, Neural Radiance Fields.

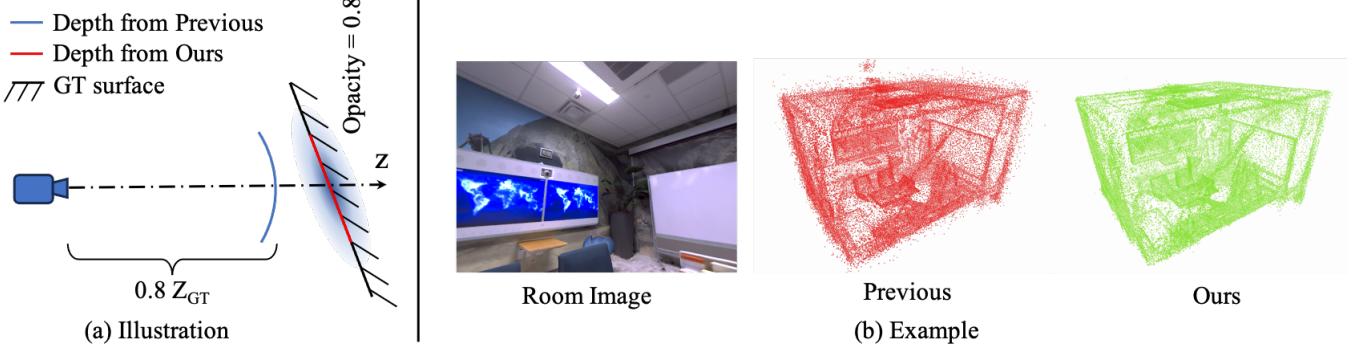


Fig. 2: **Unbiased depth rendering.** (a) Illustration of the rendered depth: We take a single Gaussian, flatten it into a plane, and fit it onto the surface as an example. Our rendered depth is the intersection point of rays and surfaces, matching the actual surface. In contrast, the depth from previous methods [11], [24] corresponds to a curved surface and may deviate from the actual surface. (b) We use true depth to supervise two different depth rendering methods. After optimization, we map the positions of all Gaussian points. Gaussians of our method fit well onto the actual surface, while the previous method results in noise and poor adherence to the surface.

## I. INTRODUCTION

**N**OVEL view synthesis and geometry reconstruction are challenging and crucial tasks in computer vision, widely used in AR/VR [13], [65], [71], 3D content generation [10], [18], [48], [53], [63], and autonomous driving. To achieve a realistic and immersive experience in AR/VR, novel view synthesis needs to be sufficiently convincing, and 3D reconstruction [32], [36], [62], [64], [66] needs to be finely detailed. Recently, neural radiance fields [22], [41], [42], [61] have been widely used to tackle this task, achieving high-fidelity novel view synthesis [2], [3], [44] and 3D geometry reconstruction [33], [56]. However, due to the computationally intensive volume rendering methods, neural radiance fields often require training times of several hours to even hundreds of hours, and rendering speeds are difficult to achieve in real-time. Recently, 3D Gaussian Splatting (3DGS) [27] has made groundbreaking advancements in this field. By optimizing the positions, rotations, scales, and appearances of the explicit 3D Gaussians and combining alpha-blend rendering, 3DGS has achieved training times in the order of minutes and rendering speeds in the millisecond range.

Although 3DGS achieves high-fidelity novel view rendering and fast training and rendering speeds. As discussed in previous methods [19], [24], Gaussians often do not conform well to actual surfaces, resulting in poor geometric accuracy. Fig. 3 also shows this conclusion. Extracting accurate meshes from millions of discrete Gaussian points is an extremely challenging task. The fundamental reason for this lies in the disorderly and irregular nature of Gaussians, which makes them unable to accurately model the surfaces of real scenes. Moreover, optimizing solely based on image reconstruction loss can easily lead to local optima, ultimately resulting in Gaussians failing to conform to actual surfaces and exhibiting poor geometric accuracy. In many practical tasks, geometric reconstruction accuracy is a crucial metric. Therefore, to address these issues, we propose a novel framework based on 3DGS that achieves high-fidelity geometric reconstruction while maintaining the high-quality rendering quality, fast training, and rendering speeds characteristic of 3DGS.

In this paper, we propose a novel unbiased depth rendering method based on 3DGS, facilitating the integration of various geometric constraints to achieve precise geometric estimation. Previous methods [24] render depth by blending the accumulations of each Gaussian at the z-position of the camera, resulting in two main issues as shown in Fig. 2. The depth corresponds to a curved surface and may deviate from the actual surface. To address these issues, we compress 3D Gaussians into flat planes and blend their accumulations to obtain normal and camera-to-plane distance maps. These maps are then transformed into depth maps. This method involves blending Gaussian plane accumulations to determine a pixel’s plane parameters. The intersection of the ray and plane defines the depth, depending on the Gaussian’s position and rotation. By dividing the distance map by the normal map, we cancel out the ray accumulation weights, ensuring the depth estimation is unbiased and falls on the estimated plane. In our experiment shown in Fig. 2, we used true depth to guide two depth rendering methods. After optimization, we mapped the positions of all Gaussian points. Results show that our method produces Gaussians that closely align with the actual surface, while the previous method generates noisy Gaussians that fail to adhere precisely to the surface.

After rendering the plane parameters for each pixel, we apply single-view and multi-view regularization to optimize these parameters. Empirically, adjacent pixels often belong to the same plane. Using this local plane assumption, we compute a normal map from neighboring pixel depth estimations and ensure consistency between this normal map and the rendered normal map. At geometric edges, the local plane assumption fails, so we detect these edges using image edges and reduce the weight in these areas, achieving smooth geometry and consistent depth and normals. However, due to the discrete and unordered nature of Gaussians, geometry may be inconsistent across multiple views. To address this, we apply multi-view regularization ensuring global geometric consistency. Similar to the Eikonal loss [56], we incorporate a multi-view geometric consistency loss to ensure smooth and consistent geometric reconstruction, even in areas with noise, blur, or weak textures.

We use two photometric coefficients to compensate for

overall changes in image brightness, further improving reconstruction quality. Finally, we validate the rendering and reconstruction quality on the MipNeRF360, the DTU [23] and the Tanks and Temples(TnT) [28] dataset. Experimental results demonstrate that, while maintaining the original Gaussian rendering quality and rendering speed, our method achieves state-of-the-art reconstruction accuracy. Moreover, our training speed only requires one hour on a single GPU, while the state-of-the-art method based on NeRF [33] requires eight GPUs over two days. In summary, our method makes the following contributions:

- We propose **a novel unbiased depth rendering method**. Based on this rendering method, we can render the reliable plane parameters for each pixel, facilitating the incorporation of various geometric constraints.
- We introduce **single-view and multi-view regularizations** to optimize the plane parameters of each pixel, achieving high-precision global geometric consistency.
- **The exposure compensation** simply and effectively enhances reconstruction accuracy.
- Our method, while maintaining the high rendering accuracy and speed of the original GS, achieves **state-of-the-art reconstruction accuracy**, and our training time is near **100 times faster** compared to state-of-the-art reconstruction methods based on NeRF [33].

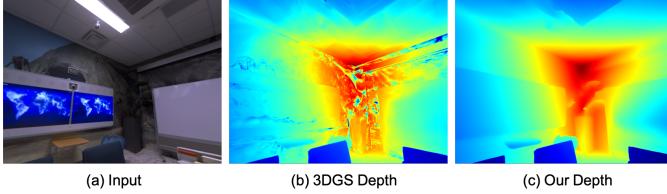


Fig. 3: **Rendered Depth.** The original depth in 3DGS exhibits significant noise, while our depth is smoother and more accurate.

## II. RELATED WORK

Surface reconstruction is a cornerstone field in computer graphics and computer vision, aimed at generating intricate and accurate surface representations from sparse or noisy input data. Obtaining high-fidelity 3D models from real-world environments is pivotal for enabling immersive experiences in augmented reality (AR) and virtual reality (VR). This paper focuses exclusively on surface reconstruction under given poses, which can be readily computed using SLAM [5], [7], [8] or SfM [43], [51], [57] methods.

### A. Traditional Surface Reconstruction

Traditional methods adhere to the universal multi-view stereo pipeline, which can be roughly categorized based on the intermediate representation they rely on, such as point cloud [16], [30], volume [29], depth map [4], [17], [52], etc. The commonly used method separates the overall MVS problem into several parts, by initially extracting dense point clouds from multi-view images through block-based matching [1], followed by the construction of surface structures

either through triangulation [6] or implicit surface fitting [25], [26]. Despite being well-established and extensively utilized in academia and industry, these traditional methods are susceptible to artifacts stemming from erroneous matching or noise introduced during the pipeline. In response, several approaches aim to enhance reconstruction completeness and accuracy by integrating deep neural networks into the matching process [50], [54].

### B. Neural Surface Reconstruction

Numerous pioneering efforts have leveraged pure deep neural networks to predict surface models directly from single or multiple image conditions using point clouds [14], [34], voxels [12], [58], and triangular meshes [32], [55] or implicit fields [40], [47] in end-to-end manner. However, these methods often incur significant computational overhead during network inference and demand extensively labeled training 3D models, hindering their real-time and real-world applicability.

With the rapid advancement in neural surface reconstruction tasks, a meticulously designed scene recovery method named NeRF [41] emerged. NeRF-based methods take 5D ray information as input and predict density and color sampled in continuous space, yielding notably more realistic rendering results. However, this representation falls short in capturing high-fidelity surfaces.

Consequently, several approaches have transformed NeRF-based network architectures into surface reconstruction frameworks by incorporating intermediate representations such as occupancy [46] or signed distance fields [56], [60]. Despite the potent surface reconstruction capabilities exhibited by NeRF-based frameworks, the stacked multi-layer-perceptron (MLP) layers impose constraints on inference time and representation ability. To address this challenge, various following studies aim to reduce dependency on MLP layers by decomposing scene information into separable structures, such as points [59] and voxels [31], [33], [35].

### C. Gaussian Splatting based Surface Reconstruction

SuGaR [19] proposed a method to extract Mesh from 3DGS. They introduced regularization terms to encourage Gaussian fitting to the scene surface. By sampling 3D point clouds from the Gaussian using the density field, they utilized Poisson reconstruction to extract a mesh from these sampled point clouds. While encouraging Gaussian fitting to the surface enhances geometric reconstruction accuracy, irregular 3D Gaussian shapes make modeling smooth geometric surfaces challenging. Moreover, due to the discreteness and disorder of the Gaussian, relying solely on image reconstruction loss can lead to overfitting, resulting in incomplete geometric information and surface mismatch. 2DGS [21] achieves view-consistent geometry by collapsing the 3D volume into a set of 2D oriented planar Gaussian disks. GOF [69] establishes a Gaussian opacity field, enabling geometry extraction by directly identifying its level-set. However, these 3DGS-based methods still produce biased depth and multi-view geometric consistency is not guaranteed. To address these issues, we flattened the Gaussian into a planar shape, which is more

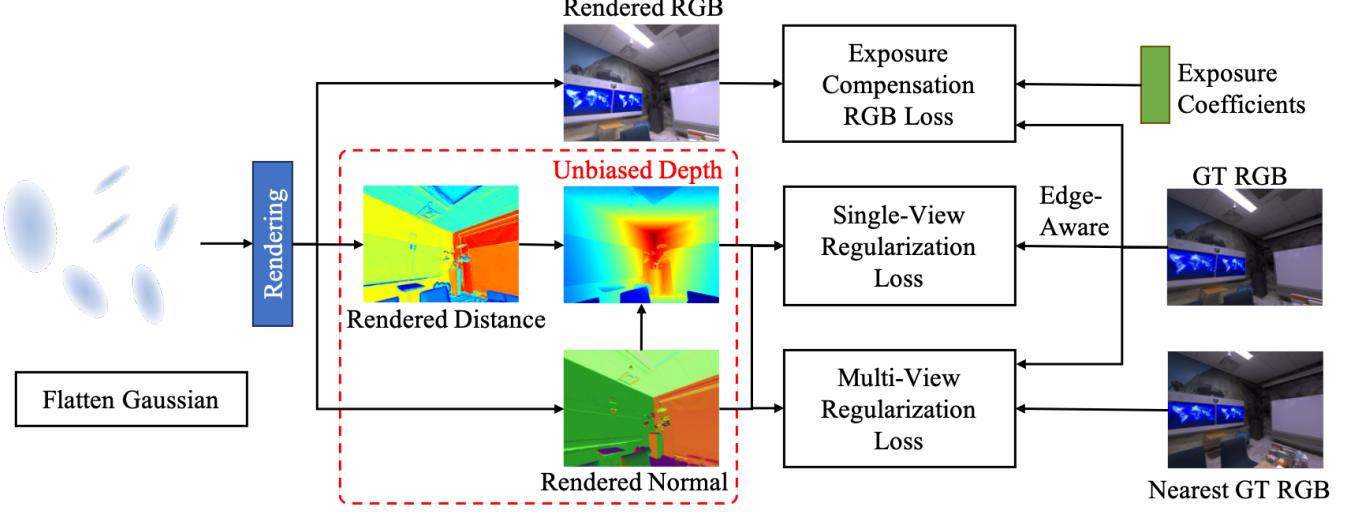


Fig. 4: **PGSR Overview.** We compress Gaussians into flat planes and render distance and normal maps, which are then transformed into unbiased depth maps. Single-view and multi-view geometric regularization ensure high precision in global geometry. Exposure compensation RGB loss enhances reconstruction accuracy.

suitable for modeling actual surfaces and facilitates rendering parameters such as normals and distances from the plane to the origin. Based on these plane parameters, we proposed unbiased depth estimation, allowing us to extract geometric parameters from the Gaussian. Then, we introduced geometric regularization terms from single-view and multi-view to optimize these geometric parameters, achieving globally consistent high-precision geometric reconstruction.

### III. PRELIMINARY OF 3D GAUSSIAN SPLATTING

3DGS [27] explicitly represents 3D scenes with a set of 3D Gaussians  $\{\mathcal{G}_i\}$ . Each Gaussian is defined by a Gaussian function:

$$\mathcal{G}_i(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)},$$

where  $\boldsymbol{\mu}_i \in \mathbb{R}^3$  and  $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3 \times 3}$  are the center of a point  $p_i \in \mathcal{P}$  and corresponding 3D covariance matrix, respectively. The covariance matrix  $\boldsymbol{\Sigma}_i$  is factorized into a scaling matrix  $\mathbf{S}_i \in \mathbb{R}^{3 \times 3}$  and a rotation matrix  $\mathbf{R}_i \in \mathbb{R}^{3 \times 3}$ :

$$\boldsymbol{\Sigma}_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^\top \mathbf{R}_i^\top.$$

3DGS allows fast  $\alpha$ -blending for rendering. Given a transformation matrix  $W$  and an intrinsic matrix  $K$ ,  $\boldsymbol{\mu}_i$  and  $\boldsymbol{\Sigma}_i$  can be transformed to camera coordinate corresponding to  $W$  and then projected to 2D coordinate:

$$\boldsymbol{\mu}'_i = \mathbf{K}W[\boldsymbol{\mu}_i, 1]^\top, \quad \boldsymbol{\Sigma}'_i = \mathbf{J}W\boldsymbol{\Sigma}_i W^\top \mathbf{J}^\top,$$

where  $J$  is the Jacobian of the affine approximation for the projective transformation. Rendering color  $\mathbf{C} \in \mathbb{R}^3$  of a pixel  $\mathbf{u}$  can be obtained in a manner of  $\alpha$ -blending:

$$\mathbf{C} = \sum_{i \in N} T_i \alpha_i \mathbf{c}_i, \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j),$$

where  $\alpha_i$  is calculated by evaluating  $\mathcal{G}_i(\mathbf{u}|\boldsymbol{\mu}'_i, \boldsymbol{\Sigma}'_i)$  multiplied with a learnable opacity corresponding to  $\mathcal{G}_i$ , and the view-dependent color  $\mathbf{c}_i \in \mathbb{R}^3$  is represented by spherical harmonics

(SH) from the Gaussian  $\mathcal{G}_i$ .  $T_i$  is the cumulative opacity.  $N$  is the number of Gaussians that the ray passes through.

The center  $\boldsymbol{\mu}_i$  of a Gaussian  $\mathcal{G}_i$  can be projected into the camera coordinate system as:

$$[x_i, y_i, z_i, 1]^\top = \mathbf{W}[\boldsymbol{\mu}_i, 1]^\top,$$

Previous Methods [11], [24] render depth under the current viewpoint:

$$D = \sum_{i \in N} T_i \alpha_i z_i.$$

### IV. METHOD

Given multi-view RGB images of static scenes, our goal is to achieve efficient and high-fidelity scene geometry reconstruction and rendering quality. Compared to 3DGS, we achieve global consistency in geometry reconstruction while maintaining similar rendering quality. Initially, we improve the modeling of scene geometry attributes by compressing 3D Gaussians into a 2D flat plane representation, which is used to generate plane distance and normal maps, and subsequently converted into unbiased depth maps. We then introduce single-view geometric, multi-view photometric, and geometric consistency loss to ensure global geometry consistency. Additionally, the exposure compensation model further improves reconstruction accuracy.

#### A. Planar-based Gaussian Splatting Representation

In this section, we will discuss how to transform 3D Gaussians into a 2D flat plane representation. Based on this plane representation, we introduce an unbiased depth rendering method, which will render plane-to-camera distance and normal maps, and can then be converted into depth maps. With geometric depth, distance, and normal maps available, it becomes easier to introduce single-view regularization and multi-view regularization in the following sections.

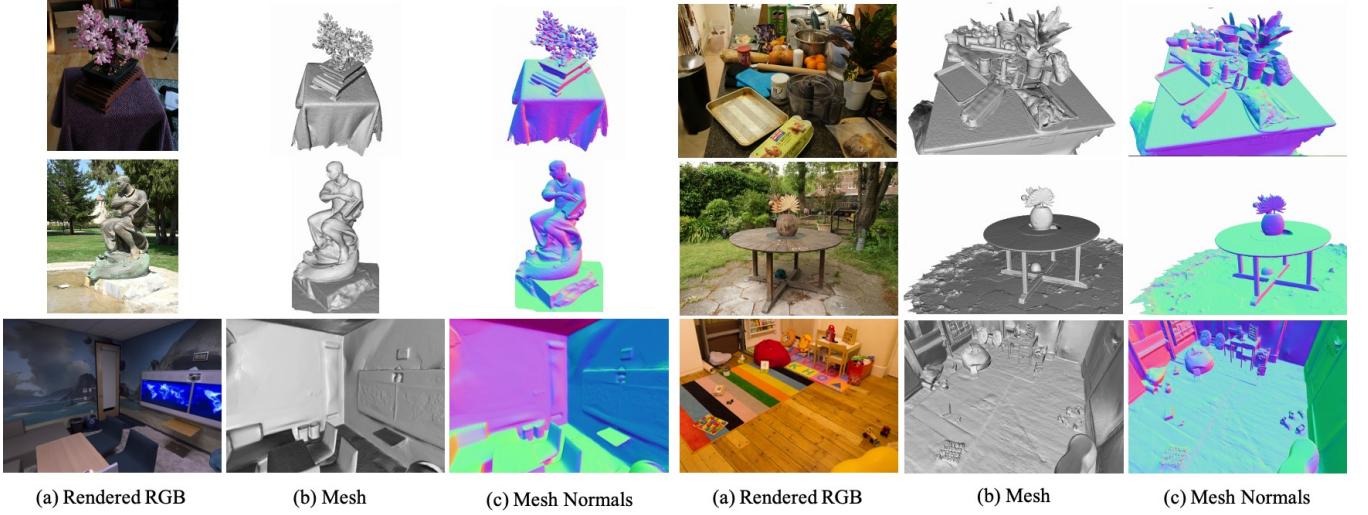


Fig. 5: **The rendering and mesh reconstruction results in various indoor and outdoor scenes that we have achieved.** PGSR achieves high-precision geometric reconstruction from a series of RGB images without requiring any prior knowledge.

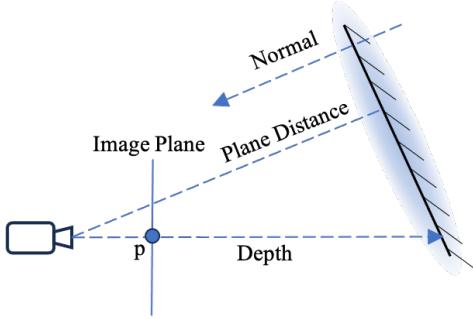


Fig. 6: **Unbiased Depth.**

Due to the difficulty in modeling real-world scene geometry attributes such as depth and normals using 3D Gaussian shapes, it's necessary to flatten the 3D Gaussians into 2D flat Gaussians in order to accurately represent the geometry surface of the actual scene. Achieving precise geometry reconstruction and high-quality rendering requires the 2D flat Gaussians to accurately conform to the scene surface. Since the 2D flat Gaussians approximate a local plane, we can conveniently render the depth and normals of the scene.

**Flattening 3D Gaussian:** The covariance matrix  $\sum_i = \mathbf{R}_i \mathbf{S}_i \mathbf{S}_i^T \mathbf{R}_i^T$  of a 3D Gaussian expresses the ellipsoidal shape. Here,  $\mathbf{R}_i$  represents the orthonormal basis of the ellipsoid's three axes, and the scale factor  $\mathbf{S}_i$  defines the size along each direction. By compressing the scale factor along specific axes, the Gaussian ellipsoid can be flattened into planes aligned with those axes. We compress the Gaussian ellipsoid along the direction of the minimum scale factor, effectively flattening the ellipsoid into a plane closest to its original shape. According to the method [9], we directly minimize the minimum scale factor  $\mathbf{S}_i = \text{diag}(s_1, s_2, s_3)$  for each Gaussian:

$$L_s = \| \min(s_1, s_2, s_3) \|_1. \quad (1)$$

**Unbiased Depth Rendering:** The direction of the minimum scale factor corresponds to the normal  $\mathbf{n}_i$  of the Gaussian. Due

to the ambiguity of the normal direction when there are two directions for the shortest axis, we resolve this issue by using the viewing direction to determine the normal direction. This implies that the angle between the viewing direction and the normal direction should be greater than 90 degrees. The final normal map under the current viewpoint is achieved through  $\alpha$ -blending:

$$\mathbf{N} = \sum_{i \in N} \mathbf{R}_c^T \mathbf{n}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (2)$$

where  $\mathbf{R}_c$  is the rotation from the camera to the global world. The distance from the plane to the camera center can be expressed as  $d_i = (\mathbf{R}_c^T(\mu_i - \mathbf{T}_c))\mathbf{R}_c^T \mathbf{n}_i^T$ , where  $\mathbf{T}_c$  is the camera center in the world.  $\mu_i$  is the center of gaussian  $G_i$ . The final distance map under the current viewpoint is achieved through  $\alpha$ -blending:

$$\mathcal{D} = \sum_{i \in N} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

Referencing Fig. 6, after obtaining the distance and normal of the plane through rendering, we can determine the corresponding depth map by intersecting rays with the plane:

$$\mathcal{D}(\mathbf{p}) = \frac{\mathcal{D}}{\mathbf{N}(\mathbf{p}) \mathbf{K}^{-1} \tilde{\mathbf{p}}}. \quad (4)$$

where  $\mathbf{p} = [u, v]^T$  is the 2D position on the image plane.  $\tilde{\mathbf{p}}$  denotes the homogeneous coordinate of  $\mathbf{p}$ , and  $\mathbf{K}$  is the intrinsic of camera.

As shown in Fig. 2, our method of rendering depth has two major advantages compared to other depth rendering techniques. First, Our depth shapes are consistent with flattened Gaussian shapes, which can truly reflect actual surfaces. Previous methods typically involve directly rendering the depth map based on  $\alpha$ -blending of the depth  $Z$  of Gaussians. Their depth is curved, inconsistent with the flat Gaussian shape, causing geometric conflicts. In contrast, we render the normal and

distance maps of the plane first and then convert them into the depth map. Our depth lies on the Gaussian fast plane. When the 3D Gaussian flat planes fit the actual surface, the rendered depth can ensure complete consistency with the actual surface. Second, since the accumulation weight for each ray may be less than 1, previous rendering methods are affected by the weight accumulation, potentially resulting in depths that are closer to the camera side and overall underestimated. In contrast, our depth is obtained by dividing the distance from the rendering origin to the plane by the normal, effectively eliminating the influence of weight accumulation coefficients.

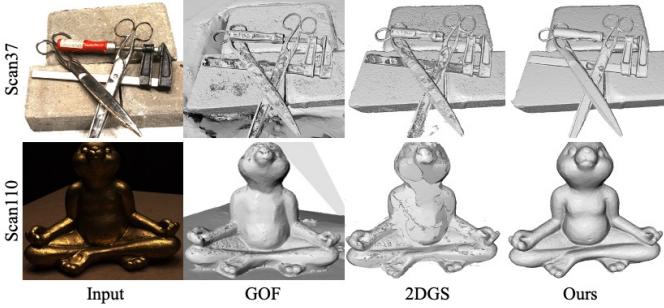


Fig. 7: **Qualitative comparison on DTU dataset.** PGSR produces smooth and detailed surfaces.

## B. Geometric Regularization

1) *Single-View Regularization*: The original 3DGS relying solely on image reconstruction loss can easily fall into local overfitting optimization, leading to Gaussian shapes inconsistent with the actual surface. Based on this, we introduce geometric constraints to ensure that the 3D Gaussian fits the actual surface as closely as possible.

**Local Plane Assumption:** Encouraged by these methods [24], [37], [49], we adopt the assumption of local planarity to constrain the local consistency of depth and normals, meaning a pixel and its neighboring pixels can be considered as an approximate plane. After rendering the depth map, we sample four neighboring points using a fixed template. With these known depths, we compute the plane’s normal. This process is repeated for the entire image, generating normals from the rendered depth map. We then minimize the difference between this normal map and the rendered normal map, ensuring geometric consistency between local depth and normals.

**Image Edge-Aware Single-View Loss:** Neighboring pixels may not necessarily fully adhere to the local planarity assumption, especially in edge regions. To address this issue, We use image edges to approximate geometric edges. For a pixel point  $\mathbf{p}$ , we sample four points from the neighboring pixels, such as up, down, left, and right. We project the four sampled depth points into 3D points  $\{\mathbf{P}_j | j = 1, \dots, 4\}$  in the camera coordinate system, then calculate the normal of the local plane for the pixel point  $\mathbf{p}$  is:

$$\mathbf{N}_d(\mathbf{p}) = \frac{(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_3 - \mathbf{P}_2)}{|(\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_3 - \mathbf{P}_2)|}, \quad (5)$$

Finally, we add the single-view normal loss is:

$$L_{svgeo} = \frac{1}{W} \sum_{\mathbf{p} \in W} |\nabla \bar{\mathbf{I}}|^5 \| \mathbf{N}_d(\mathbf{p}) - \mathbf{N}(\mathbf{p}) \|_1, \quad (6)$$

Where  $\nabla \bar{\mathbf{I}}$  is the image gradient normalized to the range of 0 to 1,  $\mathbf{N}(\mathbf{p})$  is from Equation 2, and  $W$  is the set of image pixels.

2) *Multi-View Regularization*: Single-view geometry regularization can maintain consistency between depth and normal geometry, providing fairly accurate initial geometric information. However, due to the irregular discretization of Gaussian point cloud optimization, we found that the geometry structure across multiple views is not entirely consistent. Therefore, it is necessary to introduce multi-view geometry regularization to ensure global consistency of the geometry structure.

**Multi-View Geometric Consistency**: The image loss often suffers from influences such as image noise, blur, and weak textures. In these cases, the geometric solution for photometric consistency is unreliable. Due to the discrete nature of Gaussian properties, we cannot establish a spatially dense or semi-dense SDF field as in SDF methods based on NeRF. We are unable to use spatial smoothness constraints, such as the Eikonal loss [56], to avoid the influence of unreliable solutions. To mitigate the impact of unreliable geometric solutions and ensure multi-view geometric consistency, we introduce this consistency prior constraint, which helps converge to the correct solution position, enhancing geometric smoothness.

We render the normals  $\mathbf{N}$  and the plane distances  $\mathcal{D}$  to the camera for both the reference frame and the neighboring frame. As shown in Fig. 9, for a specific pixel  $\mathbf{p}_r$  in the reference frame, the corresponding normal is  $\mathbf{n}_r$  and the distance is  $d_r$ . The pixel  $\mathbf{p}_r$  in the reference frame can be mapped to a pixel  $\mathbf{p}_n$  in the neighboring frame through the homography matrix  $\mathbf{H}_{rn}$ :

$$\tilde{\mathbf{p}}_n = \mathbf{H}_{rn}\tilde{\mathbf{p}}_r, \quad (7)$$

$$\mathbf{H}_{rn} = \mathbf{K}_n(\mathbf{R}_{rn} - \frac{\mathbf{T}_{rn}\mathbf{n}_r^T}{d_r})\mathbf{K}_r^{-1}, \quad (8)$$

where  $\tilde{\mathbf{p}}$  is the homogeneous coordinate of  $\mathbf{p}$ ,  $\mathbf{R}_{rn}$  and  $\mathbf{T}_{rn}$  are the relative transformation from the reference frame to the neighboring frame. Similarly, for the pixel  $\mathbf{p}_n$  in the neighboring frame, we can obtain the normal  $\mathbf{n}_n$  and the distance  $d_n$  to compute the homography matrix  $\mathbf{H}_{nr}$ . The pixel  $\mathbf{p}_r$  undergo forward and backward projections between the reference frame and the neighboring frame through  $\mathbf{H}_{rn}$  and  $\mathbf{H}_{nr}$ . Minimizing the forward and backward projection error constitutes the multi-view geometric consistency regularization:

$$L_{mvgeom} = \frac{1}{V} \sum_{\mathbf{p}_r \in V} \phi(\mathbf{p}_r) \quad (9)$$

where  $\phi(\mathbf{p}_r) = \| \mathbf{p}_r - \mathbf{H}_{nr}\mathbf{H}_{rn}\mathbf{p}_r \|$  is the forward and backward projection error of  $\mathbf{p}_r$ . When  $\phi(\mathbf{p}_r)$  exceeds a certain threshold, it can be considered that the pixel is occluded or that there is a significant geometric error. To prevent errors caused by occlusion, these pixels will not be included in the multi-view regularization term. If these pixels are mistakenly identified as occluded due to geometric errors, it does not

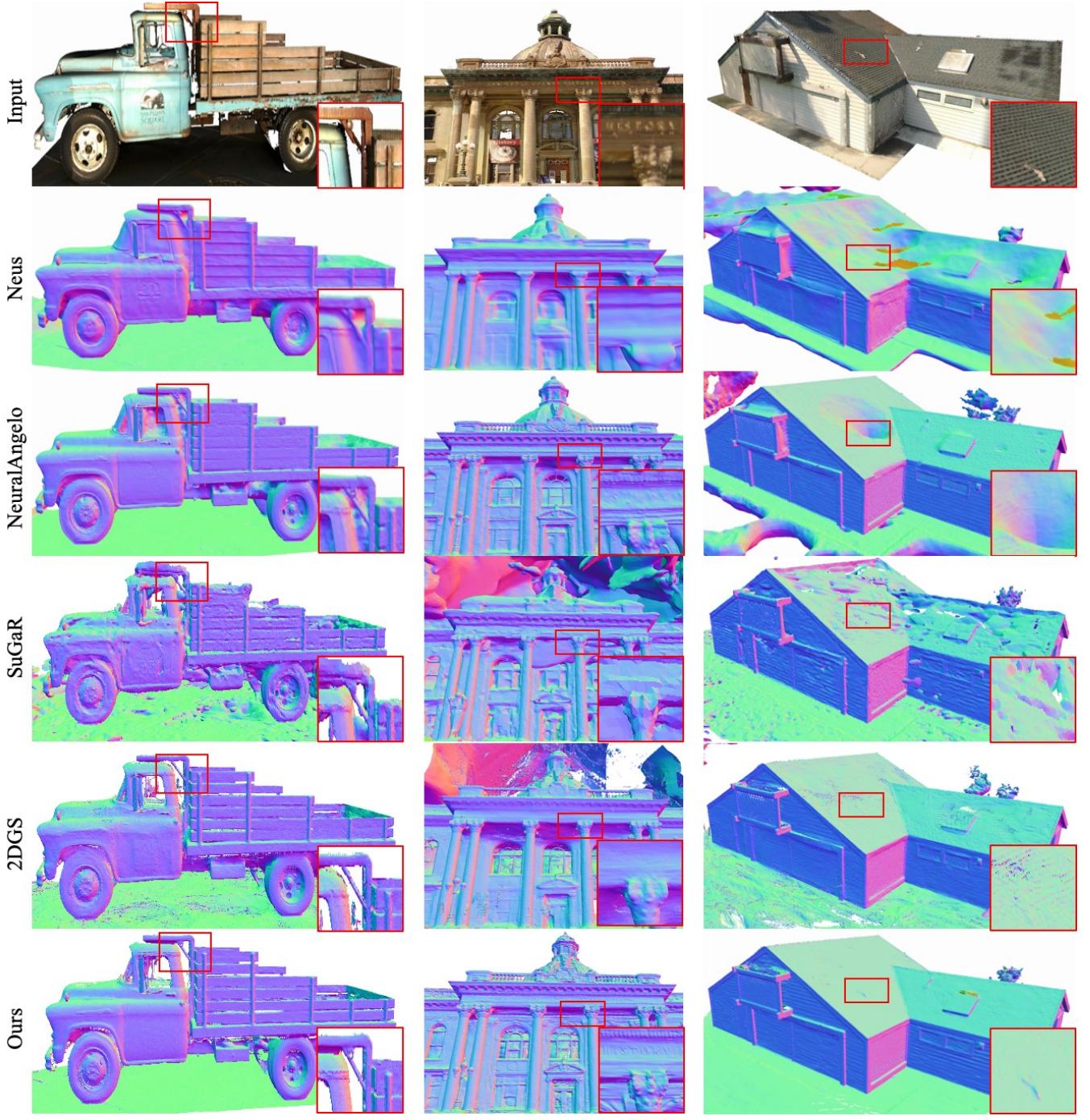


Fig. 8: **Qualitative comparison on Tanks and Temples dataset.** We visualize surface quality using a normal map generated from the reconstructed mesh. PGSR outperforms other baseline approaches in capturing scene details, whereas baseline methods exhibit missing or noisy surfaces.

affect our final convergence. This is because the single-view regularization term and the use of sparse 3D Gaussians to represent dense scenes will gradually propagate high-precision geometry, eventually leading all Gaussians to converge to the correct positions.  $V$  is the set of all pixels in the image excluding those with high forward and backward projection error.

**Multi-View Photometric Consistency:** Drawing inspiration from multi-view Stereo (MVS) methods [4], [15], [51], we employ photometric multi-view consistency constraints based on plane patches. We map a  $11 \times 11$  pixel patch  $\mathbf{P}_r$  centered at

$\mathbf{p}_r$  to the neighboring frame patch  $\mathbf{P}_n$  using the homography matrix  $\mathbf{H}_{rn}$ . Focusing on geometric details, we convert color images into grayscale. Multi-view photometric regularization requires that  $\mathbf{P}_r$  and  $\mathbf{P}_n$  should be as consistent as possible. We use the normalized cross correlation (NCC) [68] of patches in the reference frame and the neighboring frame to measure the photometric consistency:

$$\mathbf{L}_{mvrgb} = \frac{1}{V} \sum_{\mathbf{p}_r \in V} (1 - NCC(\mathbf{I}_r(\mathbf{p}_r), \mathbf{I}_n(\mathbf{H}_{rn}\mathbf{p}_r))), \quad (10)$$

where  $V$  is the set of all pixels in the image, excluding

TABLE I: Quantitative results of rendering quality for novel view synthesis on Mip-NeRF360 dataset. "Red", "Orange" and "Yellow" denote the best, second-best, and third-best results. PGSR achieves results close to 3DGS and outperforms similar reconstruction method SuGaR.

		Indoor scenes			Outdoor scenes			Average on all scenes		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF-based	NeRF [41]	26.84	0.790	0.370	21.46	0.458	0.515	24.15	0.624	0.443
	Deep Blending [20]	26.40	0.844	0.261	21.54	0.524	0.364	23.97	0.684	0.313
	INGP [44]	29.15	0.880	0.216	22.90	0.566	0.371	26.03	0.723	0.294
	M-NeRF360 [2]	31.72	0.917	0.180	24.47	0.691	0.283	28.10	0.804	0.232
	Neus [56]	25.10	0.789	0.319	21.93	0.629	0.600	23.74	0.720	0.439
GS-based	3DGS [27]	30.99	0.926	0.199	24.24	0.705	0.283	27.24	0.803	0.246
	SuGaR [19]	29.44	0.911	0.216	22.76	0.631	0.349	26.10	0.771	0.283
	2DGS [21]	30.39	0.923	0.183	24.33	0.709	0.284	27.03	0.804	0.239
	GOF [69]	30.80	0.928	0.167	24.76	0.742	0.225	27.78	0.835	0.196
	PGSR	30.41	0.930	0.161	24.45	0.730	0.224	27.43	0.830	0.193

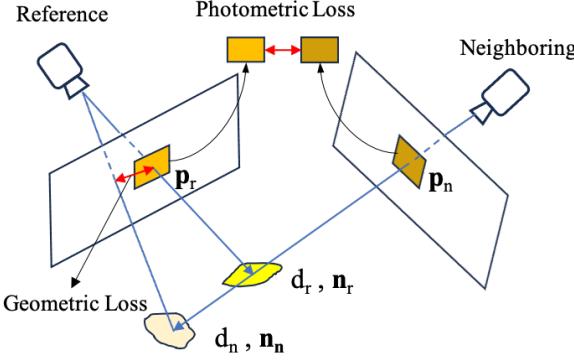


Fig. 9: Multi-view photometric and geometric loss.

those with high forward and backward projection errors.

3) *Geometric Regularization Loss*: Finally, the geometric regularization loss includes single-view geometric, multi-view geometric, and multi-view photometric consistency constraints:

$$L_{geo} = \lambda_2 L_{svgeo} + \lambda_3 L_{mvrgb} + \lambda_4 L_{mvgeom}. \quad (11)$$

### C. Exposure Compensation Image Loss

Due to changes in external lighting conditions, cameras may have different exposure times during different shooting moments, leading to overall brightness variations in images. The original 3DGS does not consider brightness changes, which can result in floating artifacts in practical scenes. To model the overall brightness variations at different times, we assign two exposure coefficients,  $a$  and  $b$ , to each image. Ultimately, images with exposure compensation can be obtained by simply computing with exposure coefficients:

$$I_i^a = \exp(a_i) I_i^r + b_i, \quad (12)$$

where  $I_i^r$  is the rendered image and  $I_i^a$  is the exposure-adjusted image. We employ the following image loss:

$$L_{rgb} = (1 - \lambda) L_1(\tilde{I} - I_i) + \lambda L_{SSIM}(I_i^r - I_i). \quad (13)$$

$$\tilde{I} = \begin{cases} I_i^a, & \text{if } L_{SSIM}(I_i^r - I_i) < 0.5 \\ I_i^r, & \text{if } L_{SSIM}(I_i^r - I_i) \geq 0.5 \end{cases} \quad (14)$$

where  $I_i$  is the ground truth image. The L1 loss constraint ensures that the exposure-adjusted image is consistent with the

ground truth image, while the SSIM loss requires the rendered image to have similar structures to the ground truth image. To enhance the robustness of exposure coefficient estimation, we need to ensure that the rendered image and the ground truth image have sufficient structural similarity before performing the estimation. After training,  $I_i^r$  is required to be globally consistent and maintain structural similarity with the ground truth image, while  $I_i^a$  can adjust the brightness of images to match the ground truth image perfectly.

### D. Training

In summary, our final training loss  $\mathbf{L}$  consists of the image reconstruction loss  $\mathbf{L}_{rgb}$ , the flattening 3D Gaussian loss  $\mathbf{L}_s$ , the geometric loss  $\mathbf{L}_{geo}$ :

$$\mathbf{L} = \mathbf{L}_{rgb} + \lambda_1 \mathbf{L}_s + \mathbf{L}_{geo}. \quad (15)$$

We set  $\lambda_1 = 100$ . For the image reconstruction loss, we set  $\lambda = 0.2$ . For the geometric loss, we set  $\lambda_2 = 0.01$ ,  $\lambda_3 = 0.2$ , and  $\lambda_4 = 0.05$ .

## V. EXPERIMENTS

**Datasets:** To validate the effectiveness of our method, we conducted experiments on various real-world datasets, including objects, and indoor and outdoor environments. We chose the widely used MiP-NeRF360 dataset [2] for evaluating novel view synthesis performance. The large and complex scenes of the TnT [28] and 15 object-centric scenes of the DTU dataset [23] were selected to assess reconstruction quality.

**Evaluation Criterion:** We chose three widely used image evaluation metrics to validate novel view synthesis: peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS) [70]. For assessing surface quality, we employed the F1 score and chamfer distance.

**Implementation Details:** Our training strategy and hyperparameters are generally consistent with 3DGS [27]. The training iterations for all scenes are set to 30,000. We adopt the densification strategy of AbsGS [67]. The learning rate for the exposure coefficient is 0.001. We begin by rendering the depth for each training view, followed by utilizing the TSDF Fusion algorithm [45] to generate the corresponding TSDF field. Subsequently, we extract the mesh [38] from the TSDF field. We only utilize the exposure compensation on the Tanks and Temples dataset. All experiments in this paper are conducted on Nvidia RTX 4090 GPU.

TABLE II: Quantitative results of chamfer distance(mm) $\downarrow$  on DTU dataset [23]. PGSR achieves the highest reconstruction accuracy and is over 100 times faster than the SDF method based on NeRF.

	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
VolSDF [60]	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	1.08	0.42	0.61	0.55	0.86	> 12h
NeuS [56]	1.00	1.37	0.93	0.43	1.10	0.65	0.57	1.48	1.09	0.83	0.52	1.20	0.35	0.49	0.54	0.84	> 12h
Neuralangelo [33]	0.37	0.72	0.35	0.35	0.87	0.54	0.53	1.29	0.97	0.73	0.47	0.74	0.32	0.41	0.43	0.61	> 128h
SuGaR [19]	1.47	1.33	1.13	0.61	2.25	1.71	1.15	1.63	1.62	1.07	0.79	2.45	0.98	0.88	0.79	1.33	1h
2DGs [21]	0.48	0.91	0.39	0.39	1.01	0.83	0.81	1.36	1.27	0.76	0.70	1.40	0.40	0.76	0.52	0.80	0.32h
GOF [69]	0.50	0.82	0.37	0.37	1.12	0.74	0.73	1.18	1.29	0.68	0.77	0.90	0.42	0.66	0.49	0.74	2h
PGSR(DS)	0.34	0.58	0.29	0.29	0.78	0.58	0.54	1.01	0.73	0.51	0.49	0.69	0.31	0.37	0.38	0.53	0.6h
PGSR	0.31	0.52	0.27	0.27	0.76	0.54	0.49	0.98	0.69	0.49	0.46	0.56	0.28	0.35	0.36	0.49	1.0h

TABLE III: Quantitative results of F1 Score $\uparrow$  for reconstruction on Tanks and Temples dataset. PGSR achieves similar reconstruction accuracy to Neuralangelo, but our training speed is over a hundred times faster.

	NeuS	Geo-Neus	Neurlangelo	SuGaR	2D GS	GOF	PGSR
Barn	0.29	0.33	0.70	0.14	0.36	0.51	0.66
Caterpillar	0.29	0.26	0.36	0.16	0.23	0.41	0.41
Courthouse	0.17	0.12	0.28	0.08	0.13	0.28	0.21
Ignatius	0.83	0.72	0.89	0.33	0.44	0.68	0.80
Meetingroom	0.24	0.20	0.32	0.15	0.16	0.28	0.29
Truck	0.45	0.45	0.48	0.26	0.26	0.58	0.60
Mean	0.38	0.35	0.50	0.19	0.30	0.46	0.50
Time	>24h	>24h	>128h	2h	34.2 m	2h	1.2h

### A. Real-time Rendering

For the validation of rendering quality, we follow the 3DGS method and conduct validation on the Mip-NeRF360 dataset [2]. We compare with current state-of-the-art methods for pure novel view synthesis as well as similar reconstruction methods to ours, including NeRF [41], Deep Blending [20], INGP [44], Mip-NeRF360 [2], NeuS [56], 3DGS [27], SuGaR [19], 2DGS [21], and GOF [69]. As shown in Table I and Fig. 5, compared to the current state-of-the-art methods, our approach not only provides excellent surface reconstruction quality but also achieves outstanding novel view synthesis results.

### B. Reconstruction

We compared our method, PGSR, with current state-of-the-art neural surface reconstruction methods including NeuS [56], Geo-NeuS [15], and NeuralAngelo [33]. We also compared it with recently emerged reconstruction methods based on 3DGS, such as SuGaR [19], 2DGS [21], and GOF [69]. All results are summarized in Fig. 5, Fig. 7, Fig. 8, Table II and Table III.

**The DTU dataset:** Our method achieves the highest reconstruction accuracy with relatively fast training speed. PGSR(DS) denotes downsampling to half the original image size for training. Our method significantly outperforms other 3DGS-based reconstruction methods. As shown in Fig. 7, our surfaces are smoother and contain more details.

**The TnT dataset:** The F1 score of PGSR is similar to NeuralAngelo and better compared to other current reconstruction methods. Our training time is over 100 times faster than NeuralAngelo. Moreover, compared to NeuralAngelo, we can reconstruct more surface details.

### C. Ablations

**Our Unbiased Depth:** From Fig 10, it can be observed that our overall geometric structure appears smoother and more

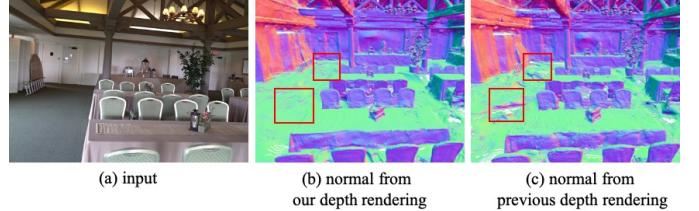


Fig. 10: The qualitative comparison of our unbiased depth method with the previous depth method [11], [24] is depicted in the normal map. Our overall geometric structure appears smoother and more precise.

TABLE IV: Ablation study on the Meetingroom of TnT dataset.

Model setting	F1-Score $\uparrow$	PSNR $\uparrow$
w/o Single-view	0.26	27.46
w/o Multi-view	0.15	28.14
w/o Our unbiased depth	0.20	26.80
Full model	0.29	27.30

precise, especially in flat regions. Table IV also demonstrates that our depth rendering method achieves higher reconstruction and rendering accuracy.

**Single-View and Multi-view Regularization:** The single-view regularization term can provide a good initial geometric accuracy without relying on multi-view information. When single-view regularization is removed, the reconstruction accuracy decreases. Multi-view regularization effectively constrains the consistency of geometry between multiple views, improving overall reconstruction accuracy. From Table IV, it is evident that multi-view regularization is crucial for reconstruction accuracy.

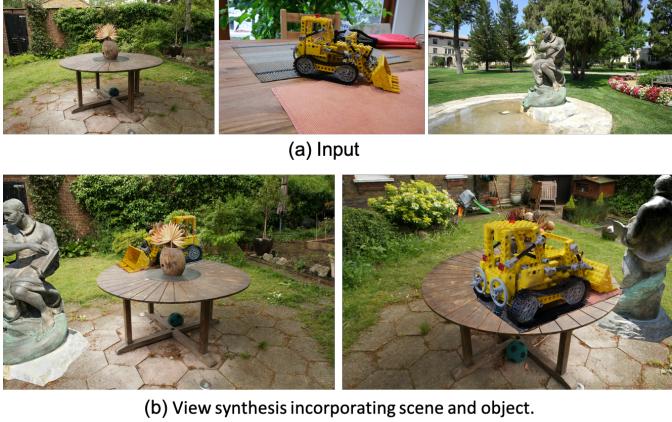
**Exposure Compensation:** We validated the exposure compensation on the Ignatius series of the TnT dataset. As shown in Table V, exposure compensation enhances reconstruction and rendering quality.

### D. Virtual Reality Application

As shown in Fig. 11, we used our method to separately reconstruct the original materials. We then extracted the excavator and Ignatius using masks and placed them in the garden scene. By rendering the scene and objects separately and using

TABLE V: Ablation study on exposure Compensation.

Model setting	F1-Score $\uparrow$	PSNR $\uparrow$
w/o exposure modeling	0.76	21.71
w exposure modeling	0.80	25.77



**Fig. 11: Virtual Reality Application.** (a) Original materials, including garden scene, excavator, and Ignatius. (b) A Virtual Reality effect showcase synthesized from these original materials.

our rendered depth to determine occlusion relationships, we achieved immersive, high-fidelity virtual reality effects with high-precision depth estimation.

## VI. LIMITATIONS AND FUTURE WORK

Although our PGSR efficiently and faithfully performs geometric reconstruction, it also faces several challenges. Firstly, we cannot perform geometric reconstruction in regions with missing or limited viewpoints, leading to incomplete or less accurate geometry. Exploring methods to improve reconstruction quality under insufficient constraints using priors is another avenue for further investigation. Secondly, our method does not consider scenarios involving reflective surfaces or mirrors, so reconstruction in these environments will pose challenges. Integrating with existing 3DGS work that accounts for reflective surfaces would enhance reconstruction accuracy in such scenarios. Finally, we found that there are some floating points in the scene, which affect the rendering and reconstruction quality. Integrating more advanced 3DGS baselines [39] would help further enhance overall quality.

## VII. CONCLUSION

In this paper, we propose a novel unbiased depth rendering method based on 3DGS. With this method, we render the plane geometry parameters for each pixel, including normal, distance, and depth maps. We then incorporate single-view and multi-view geometric regularization, and exposure compensation model to achieve precise global consistency in geometry. We validate our rendering and reconstruction quality on the MipNeRF360, DTU, and TnT datasets. The experimental results indicate that our method achieves the highest geometric reconstruction accuracy and rendering quality compared to the current state-of-the-art methods.

## REFERENCES

- [1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.
- [2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.
- [3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.
- [4] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12–18, 2008, Proceedings, Part I 10*, pages 766–779. Springer, 2008.
- [5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.
- [6] Frédéric Cazals and Joachim Giesen. Delaunay triangulation based surface reconstruction. In *Effective computational geometry for curves and surfaces*, pages 231–276. Springer, 2006.
- [7] Danpeng Chen, Nan Wang, Runsen Xu, Weijian Xie, Hujun Bao, and Guofeng Zhang. Rnn-vio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 275–283. IEEE, 2021.
- [8] Danpeng Chen, Shuai Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Hujun Bao, and Guofeng Zhang. Vip-slam: An efficient tightly-coupled rgb-d visual inertial planar slam. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5615–5621. IEEE, 2022.
- [9] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023.
- [10] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, and Chi Zhang. Artist-Created Mesh Generation with Autoregressive Transformers. *arXiv*, 2024.
- [11] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. *arXiv preprint arXiv:2402.14650*, 2024.
- [12] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *European Conference on Computer Vision*, volume 9912, pages 628–644, 2016.
- [13] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praeneeth Chakravarthula, Xubo Yang, and Qi Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3854–3864, 2022.
- [14] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2463–2471, 2017.
- [15] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geonus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35:3403–3416, 2022.
- [16] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.
- [17] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2015.
- [18] Peng Gao, Le Zhuo, Dongyang Liu, Ruoyi Du, Xu Luo, Longtian Qiu, Yuhang Zhang, Chen Lin, Rongjie Huang, Shijie Geng, Renrui Zhang, Junlin Xi, Wenqi Shao, Zhengkai Jiang, Tianshuo Yang, Weicai Ye, He Tong, Jingwen He, Yu Qiao, and Hongsheng Li. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arxiv:2405.05945*, 2024.
- [19] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. *arXiv preprint arXiv:2311.12775*, 2023.
- [20] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

- [21] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *arXiv preprint arXiv:2403.17888*, 2024.
- [22] Chenxi Huang, Yuenan Hou, Weicai Ye, Di Huang, Xiaoshui Huang, Binbin Lin, Deng Cai, and Wanli Ouyang. Nerf-det++: Incorporating semantic cues and perspective-aware depth supervision for indoor multi-view 3d detection. *arXiv preprint arXiv:2402.14464*, 2024.
- [23] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.
- [24] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023.
- [25] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.
- [26] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.
- [27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.
- [28] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.
- [29] Kiryakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000.
- [30] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005.
- [31] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-surf: Voxel-based implicit surface representation. *IEEE Transactions on Visualization and Computer Graphics*, 2022.
- [32] Hai Li, Weicai Ye, Guofeng Zhang, Sanyuan Zhang, and Hujun Bao. Saliency guided subdivision for single-view mesh reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 1098–1107. IEEE, 2020.
- [33] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8465, 2023.
- [34] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Conference on Artificial Intelligence*, pages 7114–7121, 2018.
- [35] Lingjie Liu, Jitao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems*, pages 15651–15663, 2020.
- [36] Xiangyu Liu, Weicai Ye, Chaoran Tian, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Coxgraph: multi-robot collaborative, globally consistent, online dense reconstruction system. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8722–8728. IEEE, 2021.
- [37] Xiaoxiao Long, Yuhang Zheng, Yupeng Zheng, Beiwen Tian, Cheng Lin, Lingjie Liu, Hao Zhao, Guyue Zhou, and Wenping Wang. Adaptive surface normal constraint for geometric estimation from monocular images. *arXiv preprint arXiv:2402.05869*, 2024.
- [38] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998.
- [39] Tao Lu, Mulin Yu, Lining Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. *arXiv preprint arXiv:2312.00109*, 2023.
- [40] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.
- [41] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- [42] Yuhang Ming, Weicai Ye, and Andrew Calway. idf-slam: End-to-end rgb-d slam with neural implicit mapping and deep feature tracking. *arXiv preprint arXiv:2209.07919*, 2022.
- [43] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. In *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*, pages 257–270. Springer Berlin Heidelberg, 2012.
- [44] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.
- [45] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.
- [46] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3501–3512, 2020.
- [47] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.
- [48] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [49] Xiaojuan Qi, Renjie Liao, Zhengze Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018.
- [50] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019.
- [51] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [52] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [53] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.
- [54] Fangjinhuai Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14194–14203, 2021.
- [55] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single RGB images. In *European Conference on Computer Vision*, volume 11215, pages 55–71, 2018.
- [56] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
- [57] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013.
- [58] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2Vox: Context-aware 3d reconstruction from single and multi-view images. In *IEEE/CVF International Conference on Computer Vision*, pages 2690–2698, 2019.
- [59] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022.
- [60] Lior Yariv, Jitao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*, pages 4805–4815, 2021.
- [61] Weicai Ye, Shuo Chen, Chong Bao, Hujun Bao, Marc Pollefeys, Zhaopeng Cui, and Guofeng Zhang. IntrinsicNeRF: Learning Intrinsic Neural Radiance Fields for Editable Novel View Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.
- [62] Weicai Ye, Xinyu Chen, Ruohao Zhan, Di Huang, Xiaoshui Huang, Haoyi Zhu, Hujun Bao, Wanli Ouyang, Tong He, and Guofeng Zhang. Dynamic-Aware Tracking Any Point for Structure from Motion in the Wild. *arXiv preprint*, 2024.
- [63] Weicai Ye, Chenhao Ji, Zheng Chen, Junyao Gao, Xiaoshui Huang, Song-Hai Zhang, Wanli Ouyang, Tong He, Cairong Zhao, and Guofeng Zhang. DiffPano: Scalable and Consistent Text to Panorama Generation with Spherical Epipolar-Aware Diffusion. *arXiv preprint*, 2024.

- [64] Weicai Ye, Xinyue Lan, Shuo Chen, Yuhang Ming, Xingyuan Yu, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Pvo: Panoptic visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9579–9589, June 2023.
- [65] Weicai Ye, Hai Li, Tianxiang Zhang, Xiaowei Zhou, Hujun Bao, and Guofeng Zhang. SuperPlane: 3D plane detection and description from a single image. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 207–215. IEEE, 2021.
- [66] Weicai Ye, Xingyuan Yu, Xinyue Lan, Yuhang Ming, Jinyu Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Deflowslam: Self-supervised scene motion decomposition for dynamic dense slam. *arXiv preprint arXiv:2207.08794*, 2022.
- [67] Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. Absgs: Recovering fine details for 3d gaussian splatting. *arXiv preprint arXiv:2404.10484*, 2024.
- [68] Jae-Chern Yoo and Tae Hee Han. Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28:819–843, 2009.
- [69] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024.
- [70] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [71] Tianxiang Zhang, Chong Bao, Hongjia Zhai, Jiazen Xia, Weicai Ye, and Guofeng Zhang. Arcargo: Multi-device integrated cargo loading management system with augmented reality. In *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 341–348. IEEE, 2021.

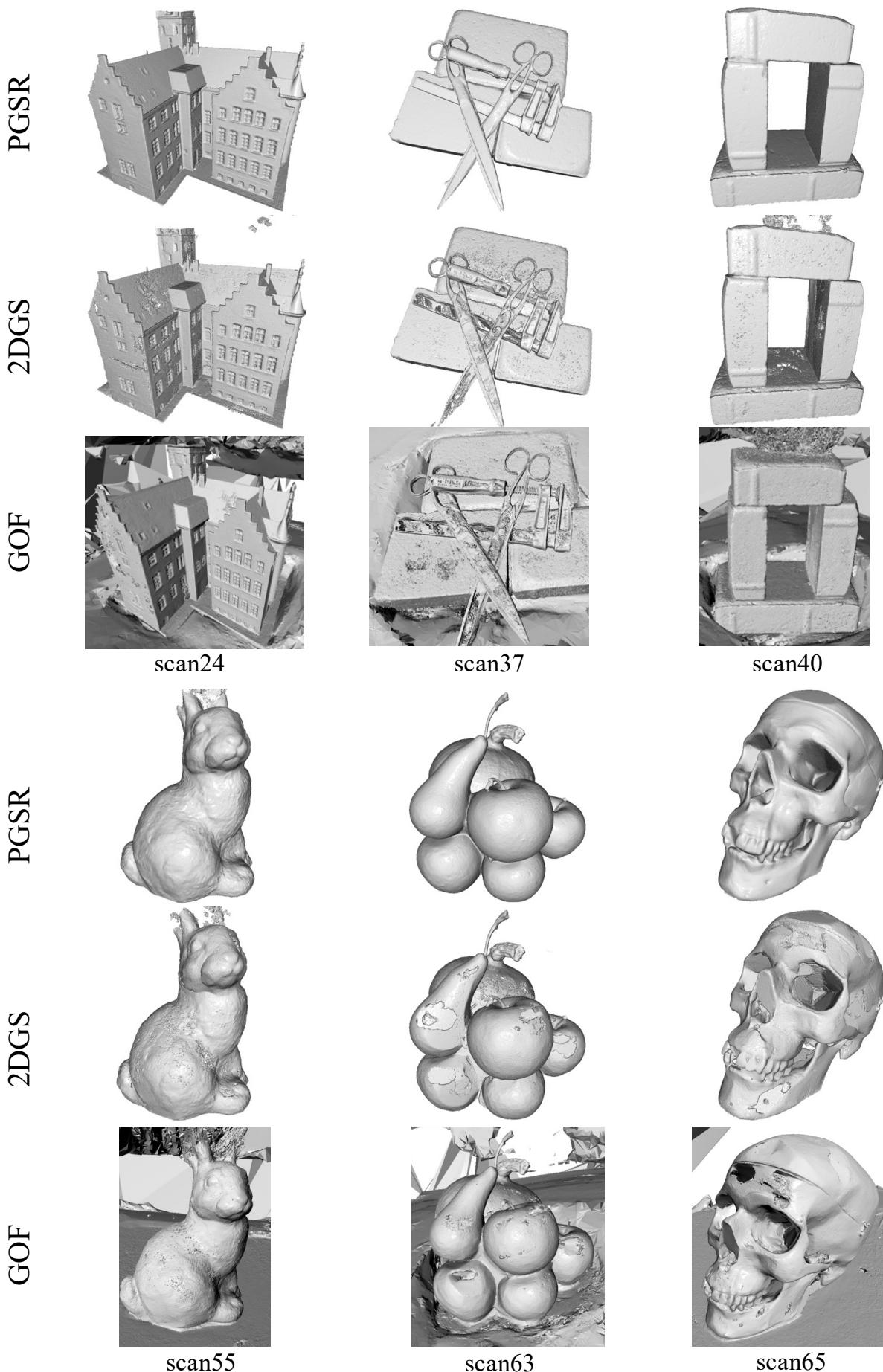


Fig. 12: Qualitative comparisons in surface reconstruction between PGSR, 2DGS, and GOF on the DTU dataset.

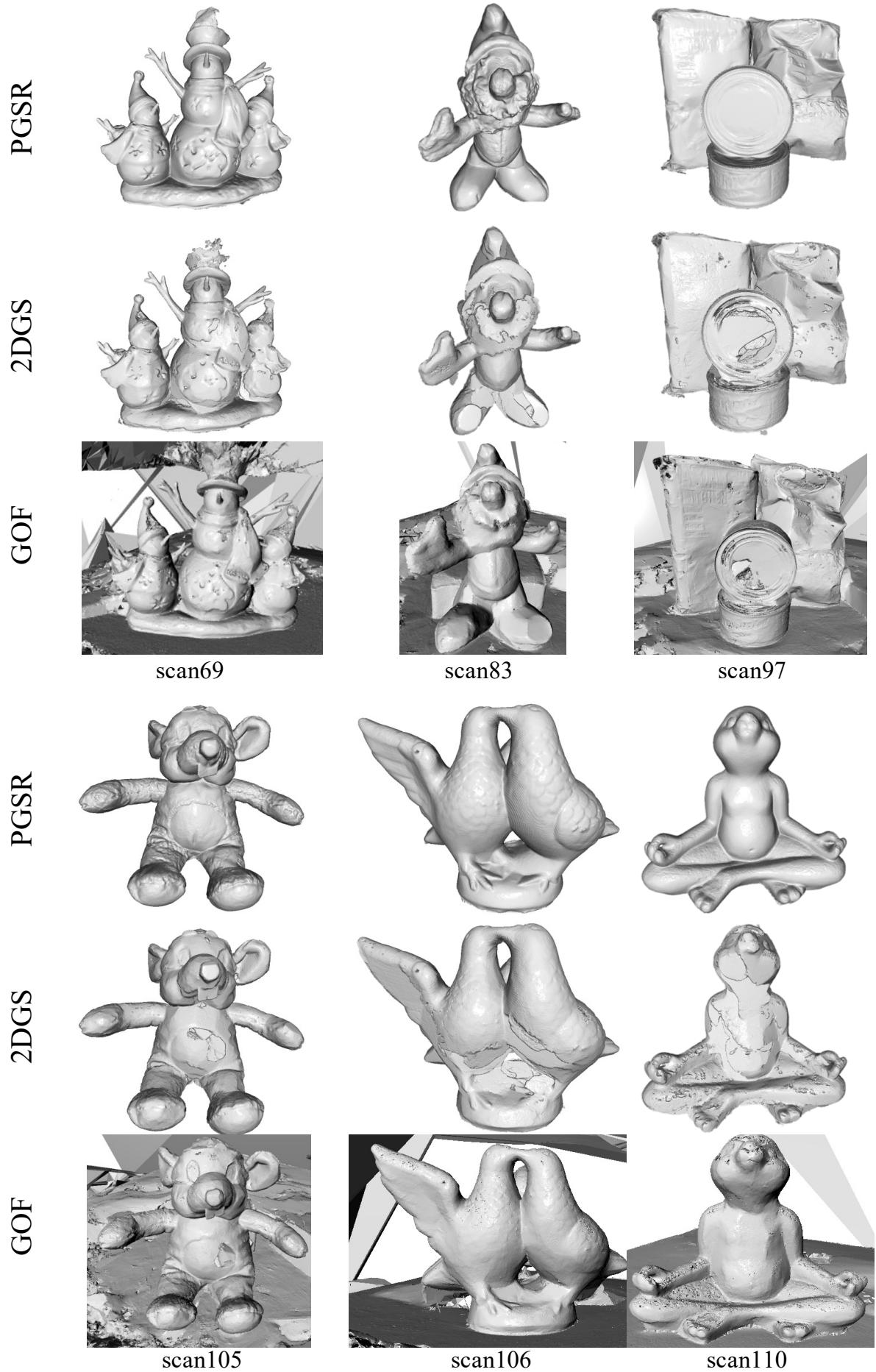


Fig. 13: Qualitative comparisons in surface reconstruction between PGSR, 2DGs, and GOF on the DTU dataset.

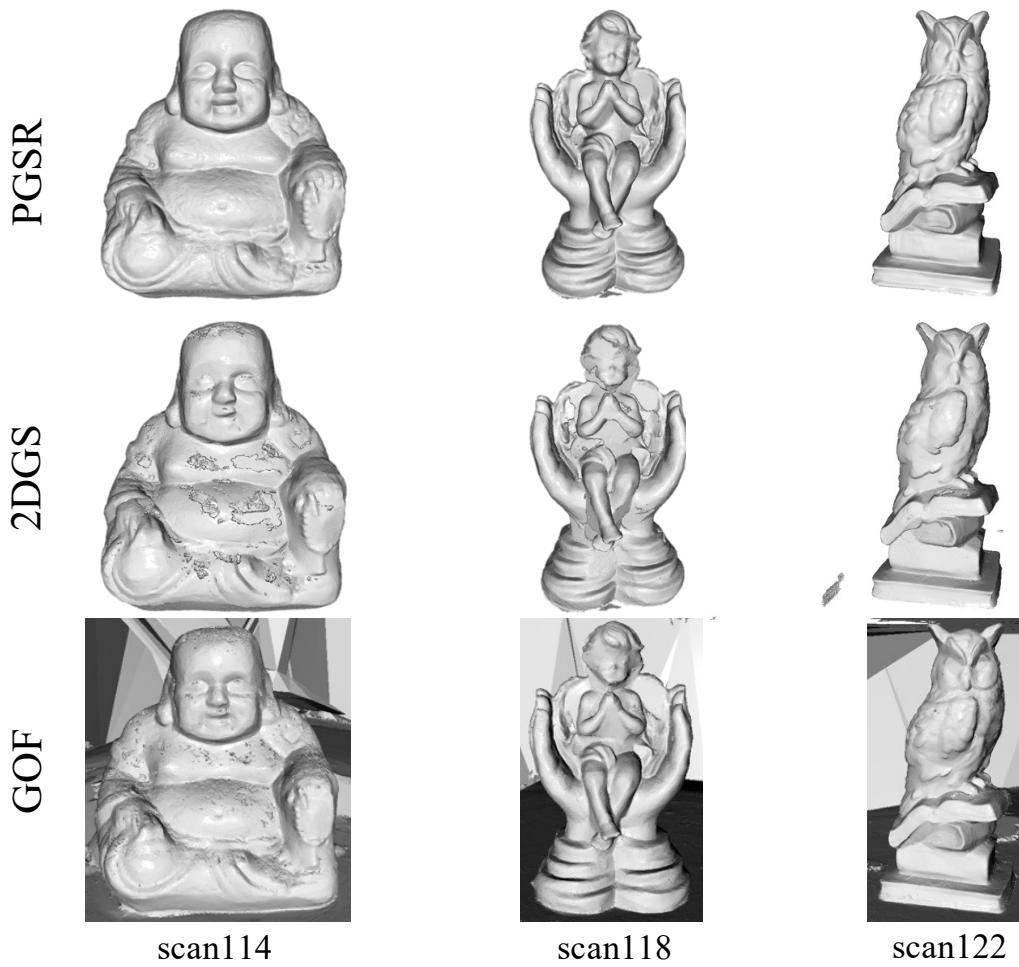


Fig. 14: Qualitative comparisons in surface reconstruction between PGSR, 2DGS, and GOF on the DTU dataset.



Fig. 15: Qualitative comparisons in surface reconstruction between PGSR, 2DGS, and GOF.

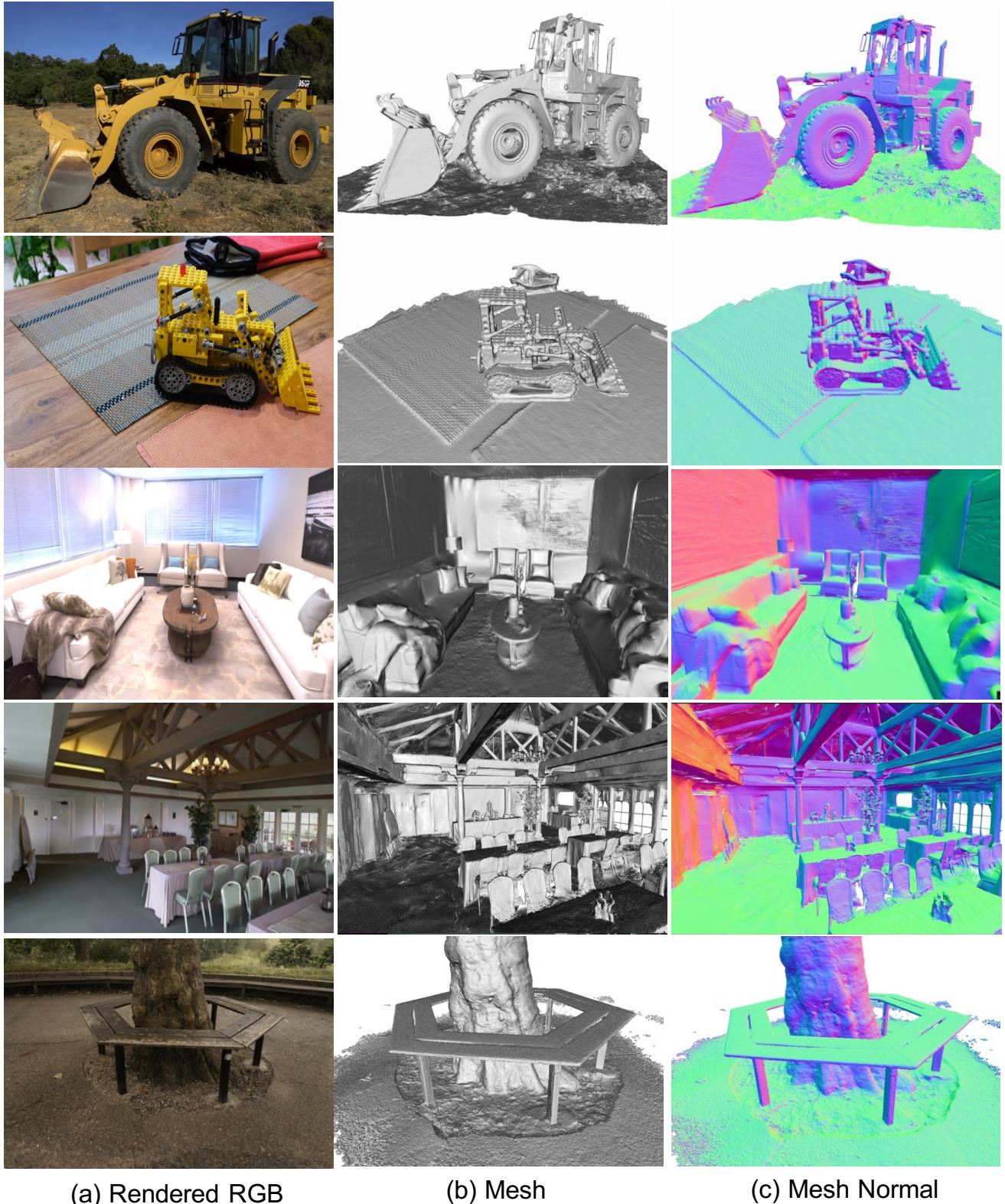


Fig. 16: PGSR achieves high-precision geometric reconstruction in various indoor and outdoor scenes from a series of RGB images without requiring any prior knowledge.