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

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, *Graduate Student Member, IEEE*, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, *Member, IEEE*, and Guofeng Zhang, *Member, IEEE*

***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 representation (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 the proposed method achieves fast training and rendering while maintaining high-fidelity rendering and geometric reconstruction, outperforming 3DGS-based and NeRF-based methods.**

***Index Terms*—Neural radiance fields, neural rendering, planar-based Gaussian splatting, surface reconstruction.**

Received 14 June 2024; revised 24 September 2024; accepted 3 November 2024. Date of publication 7 November 2024; date of current version 1 August 2025. This work was supported in part the by NSF of China under Grant 61932003. Recommended for acceptance by C. Li. *(Corresponding author: Guofeng Zhang.)*

Danpeng Chen is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China, and also with Tetras.AI, Shenzhen 518066, China (e-mail: 11921155@zju.edu.cn).

Hai Li is with RayNeo, Shenzhen, Guangdong 518000, China (e-mail: lihai@ffalcon.cn).

Weicai Ye, Hujun Bao, and Guofeng Zhang are with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China (e-mail: maikeyeweicai@gmail.com; baohujun@zju.edu.cn; zhangguofeng@zju.edu.cn).

Yifan Wang is with Shanghai AI Laboratory, Shanghai 200231, China (e-mail: wangyifan@pjlab.org.cn).

Weijian Xie is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China, and also with SenseTime Research, Hangzhou 311215, China (e-mail: xieweijian@sensetime.com).

Shangjin Zhai, Nan Wang, and Haomin Liu are with SenseTime Research, Hangzhou 311215, China (e-mail: zhaishangjin@sensetime.com; wangnan@sensetime.com; liuhaomin@sensetime.com).

Our code will be made publicly available, and more information can be found on our project page (https://zju3dv.github.io/pgsr/).

This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2024.3494046, provided by the authors.

Digital Object Identifier 10.1109/TVCG.2024.3494046

I. Introduction

NOVEL view synthesis and geometry reconstruction are challenging and crucial tasks in computer vision, widely used in AR/VR [1], [2], 3D content generation [3], [4], [5], [6], and autonomous driving. To achieve a realistic and immersive experience in AR/VR, novel view synthesis needs to be sufficiently convincing, and 3D reconstruction [7], [8], [9], [10] needs to be finely detailed. Recently, neural radiance fields [11], [12], [13], [14] have been widely used to tackle this task, achieving high-fidelity novel view synthesis [15], [16], [17] and 3D geometry reconstruction [18], [19]. 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) [20] 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 [21], [22], 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 [22] 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

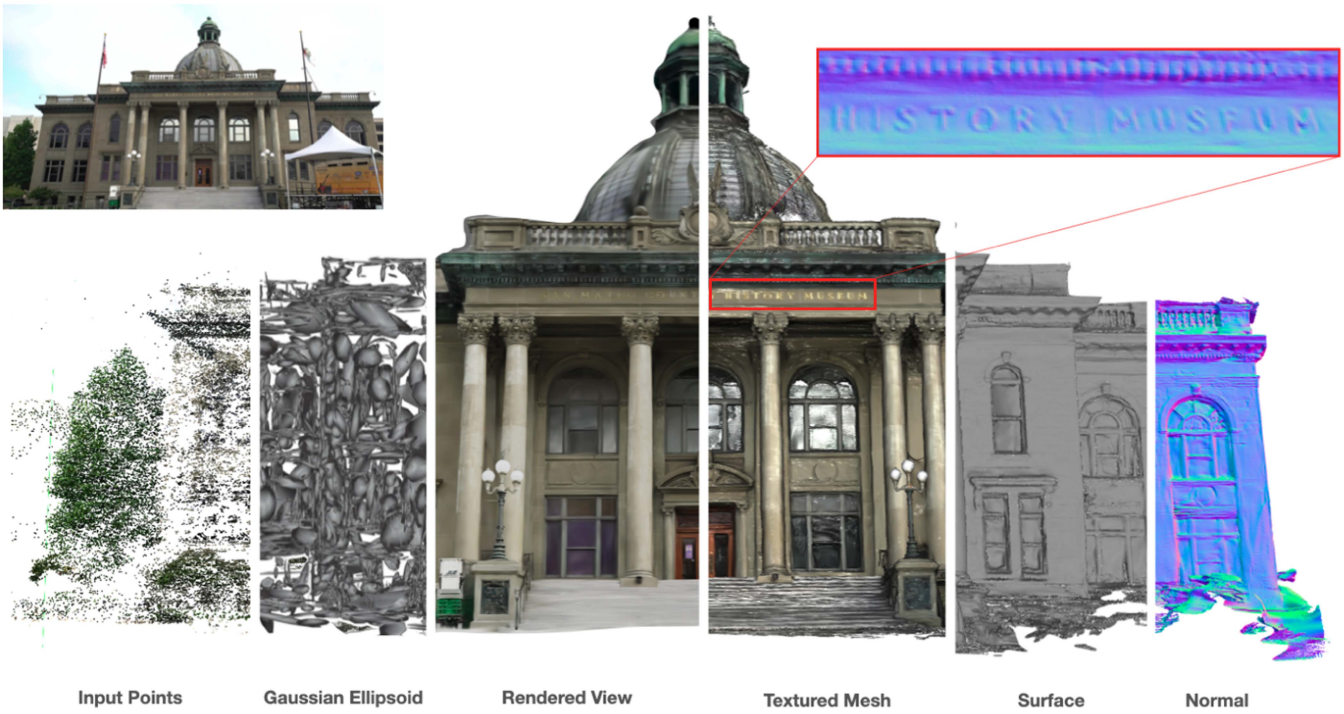

Fig. 1. PGSR representation. We present a Planar-based Gaussian Splatting Reconstruction representation for efficient and high-fidelity surface reconstruction from multi-view RGB images without any geometric prior (depth or normal from pre-trained model). The courthouse reconstructed by our method demonstrates 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.

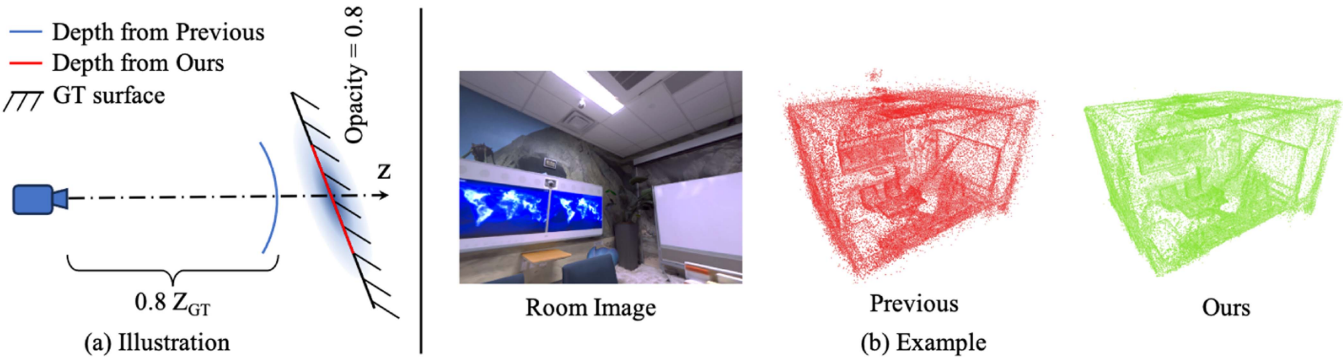

Fig. 2. Unbiased depth rendering. (a) Illustration of the rendered depth: We take a single Gaussian, flatten it into a plane, and fit it onto the surface as an example. Our rendered depth is the intersection point of rays and surfaces, matching the actual surface. In contrast, the depth from previous methods [22], [68] 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.

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

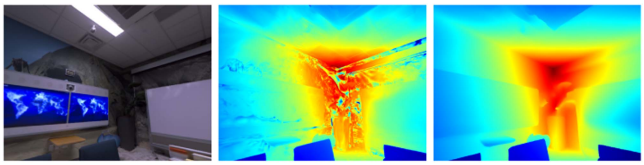

Fig. 3. Rendered Depth. The original depth in 3DGS exhibits significant noise, while our depth is smoother and more accurate.

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 [19], we incorporate a multi-view geometric consistency loss to ensures 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) [24] 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 [18] 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 [18].

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 [25], [26], [27] or SFM [28], [29], [30] 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 [31], [32], volume [33], depth map [34], [35], [36], 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 [37], followed by the construction of surface structures either through triangulation [38] or implicit surface fitting [39], [40]. 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 [41], [42].

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 [43], [44], voxels [45], [46], and triangular meshes [7], [47] or implicit fields [48], [49] 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 [11] 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 [50] or signed distance fields [19], [51]. 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 [52] and voxels [18], [53], [54].

C. Gaussian Splatting Based Surface Reconstruction

SuGaR [21] 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. However, biased depth is used to constrain the density field, with the aim of extracting surface points from the density field. The final surface quality depends on the depth quality, and it is difficult to reconstruct smooth surfaces from a discrete density field. Due to the discreteness and randomness of Gaussian points, relying solely on image reconstruction constraints without proper geometric regularization can easily result in local optimization, making it difficult to reconstruct high-precision surfaces. While our method shares some conceptual similarities with SuGaR,

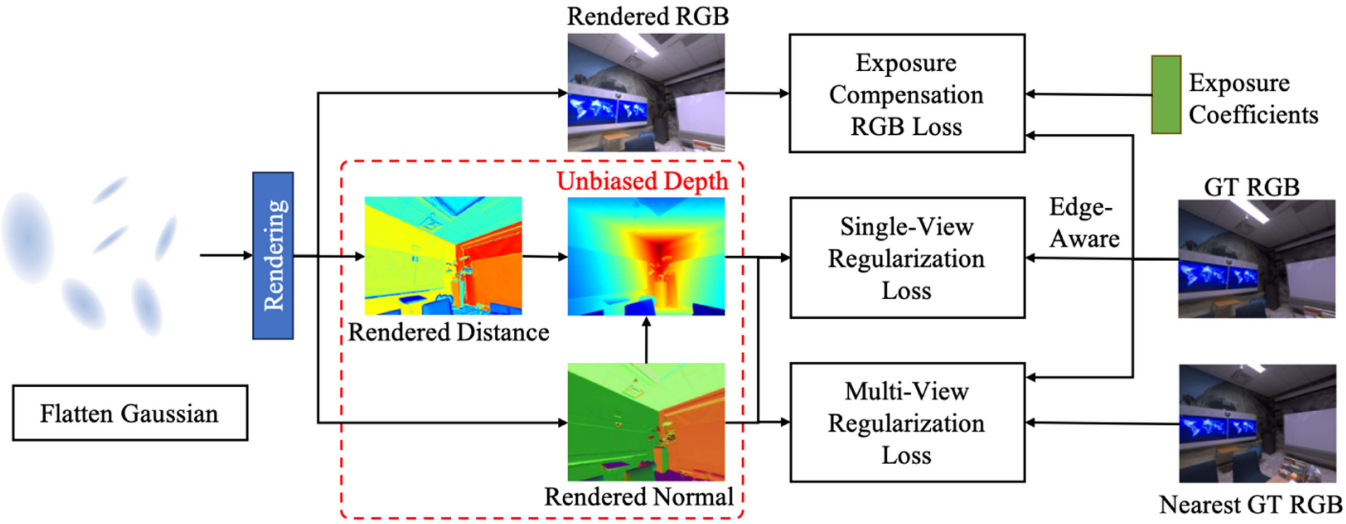

Fig. 4. PGSR Overview. We compress Gaussians into flat planes and render distance and normal maps, which are then transformed into unbiased depth maps. Single-view and multi-view geometric regularization ensure high precision in global geometry. Exposure compensation RGB loss enhances reconstruction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

such as approximating 3D Gaussian ellipsoids as planes, using the shortest axis as the plane normal representation, and aiming to represent actual surfaces with planes, there are significant differences in the plane rendering method and the use of planes. The concurrent works that are very close in time to ours are 2DGS [55] and GOF [56]. 2DGS achieves consistent geometry across views by collapsing the 3D volume into a collection of 2D oriented planar Gaussian disks. GOF forms a Gaussian opacity field, facilitating geometry extraction by directly identifying its level set. However, these Gaussian splatting-based methods still fail to produce high-precision depth and cannot ensure multi-view geometric consistency. 2DGS uses planes to resolve the 3D Gaussian geometric ambiguity in multi-view scenarios. It uses two depth rendering methods, requiring manual selection between the median and expected depth value of ray-plane intersections. In boundary scenarios, 2DGS recommends using median depth. However, median depth suffers from the issue of 'disk-aliasing'. Additionally, there are no constraints to ensure multi-view consistency. To address these issues, we flattened the Gaussian into a planar shape, which is more 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 [20] explicitly represents 3D scenes with a set of 3D Gaussians $\{\mathcal{G}_i\}$. Each Gaussian is defined by a Gaussian function:

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

where $\boldsymbol{\mu}_i\in\mathbb{R}^3$ and $\boldsymbol{\Sigma}_i\in\mathbb{R}^{3\times3}$ are the center of a point $\boldsymbol{p}_i\in\mathcal{P}$ and corresponding 3D covariance matrix, respectively. The covariance matrix $\boldsymbol{\Sigma}_i$ can be decomposed into a scaling matrix $\boldsymbol{S}_i\in\mathbb{R}^{3\times3}$ and a rotation matrix $\boldsymbol{R}_i\in\mathbb{R}^{3\times3}$:

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

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

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

where J denotes the Jacobian matrix of the projective transformation. Rendering color $\boldsymbol{C}\in\mathbb{R}^3$ of a pixel $\boldsymbol{u}$ can be obtained in a manner of α-blending:

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

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

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

Previous Methods [22], [68] render depth under the current viewpoint:

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

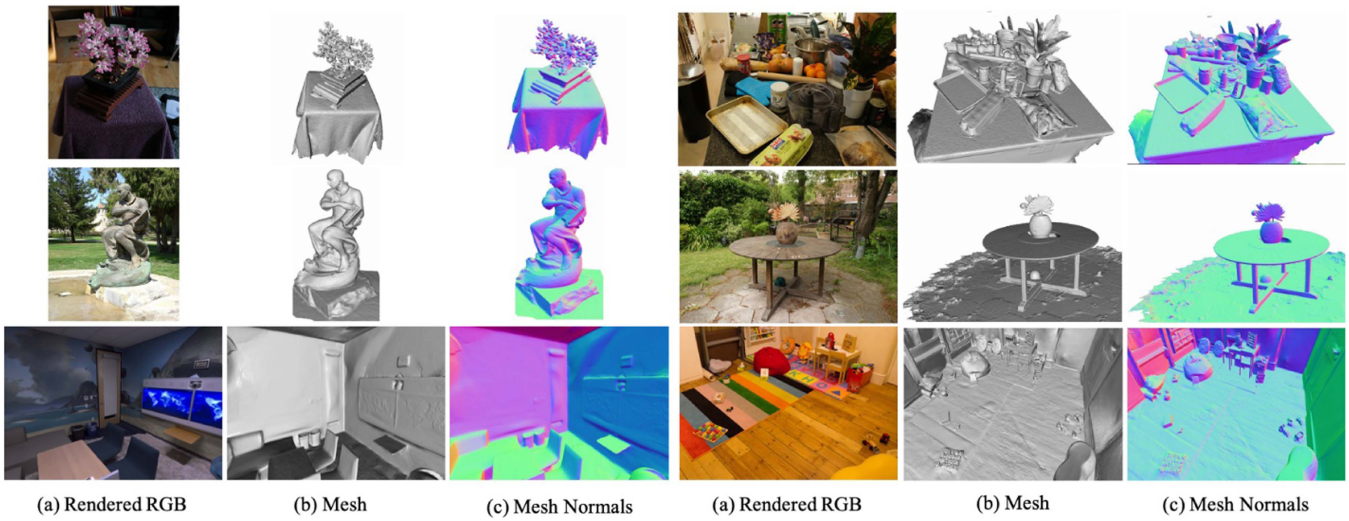

Fig. 5. The rendering and mesh reconstruction results in various indoor and outdoor scenes that we have achieved. PGSR achieves high-precision geometric reconstruction from a series of RGB images without requiring any prior knowledge.

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.

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 $\sum_i = \boldsymbol{R}_i \boldsymbol{S}_i \boldsymbol{S}_i^T \boldsymbol{R}_i^T$ of a 3D Gaussian expresses the ellipsoidal shape. Here, $\boldsymbol{R}_i$ represents the orthonormal basis of the ellipsoid's three axes, and the scale factor $\boldsymbol{S}_i$ 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 [57], we directly minimize the minimum scale factor $\boldsymbol{S}_i = \mathrm{diag}(s_1, s_2, s_3)$ for each Gaussian:

$$L_s = \| \min(s_1, s_2, s_3) \|_1 . \tag{1}$$

Unbiased Depth Rendering: The direction of the minimum scale factor corresponds to the normal $\boldsymbol{n}_i$ 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 α-blending:

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

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

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

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:

$$\boldsymbol{D}(\boldsymbol{p}) = \frac{\boldsymbol{\mathcal{D}}}{\boldsymbol{N}(p) \boldsymbol{K}^{-1} \tilde{\boldsymbol{p}}}, \tag{4}$$

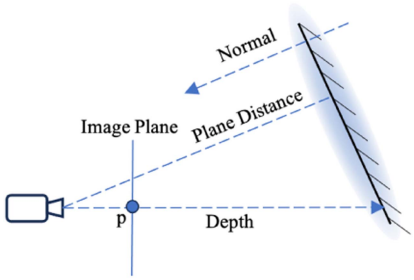

Fig. 6. Unbiased Depth.

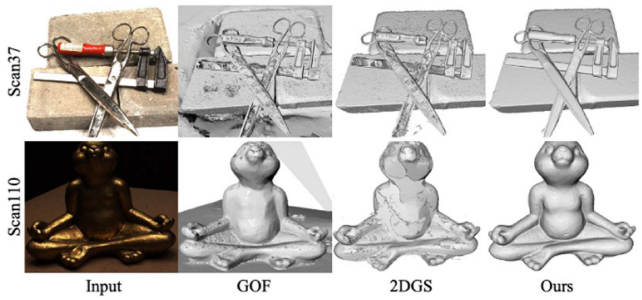

Fig. 7. Qualitative comparison on DTU dataset. PGSR produces smooth and detailed surfaces.

where $\boldsymbol{p} = [u, v]^T$ indicates the 2D position on the image plane. $\tilde{\boldsymbol{p}}$ is the homogeneous coordinate representation of $\boldsymbol{p}$, and $\boldsymbol{K}$ refers to 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 α-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 flat 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.

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 [22], [58], [59], 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 $\boldsymbol{p}$, 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 $\{\boldsymbol{P}_j | j = 1, \ldots, 4\}$ in the camera coordinate system, then calculate the normal of the local plane for the pixel point $\boldsymbol{p}$ is:

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

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

$$\boldsymbol{L}_{svgeom} = \frac{1}{W} \sum_{\boldsymbol{p} \in W} \left(1 - \overline{\nabla \boldsymbol{I}}\right)^2 \| \boldsymbol{N}_d(\boldsymbol{p}) - \boldsymbol{N}(\boldsymbol{p}) \|_1, \tag{6}$$

where $\overline{\nabla \boldsymbol{I}}$ is the image gradient normalized to the range of 0 to 1, $\boldsymbol{N}(\boldsymbol{p})$ is from (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 [19], 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 $\boldsymbol{N}$ and the plane distances $\boldsymbol{\mathcal{D}}$ to the camera for both the reference frame and the neighboring frame. As shown in Fig. 9, for a specific pixel $\boldsymbol{p}_r$ in the reference frame, the corresponding normal is $\boldsymbol{n}_r$ and the distance is d_r. The pixel $\boldsymbol{p}_r$ in the reference frame can be mapped to a pixel $\boldsymbol{p}_n$ in the neighboring frame through the homography matrix $\boldsymbol{H}_{rn}$:

$$\tilde{\boldsymbol{p}_n} = \boldsymbol{H}_{rn} \tilde{\boldsymbol{p}_r}, \tag{7}$$

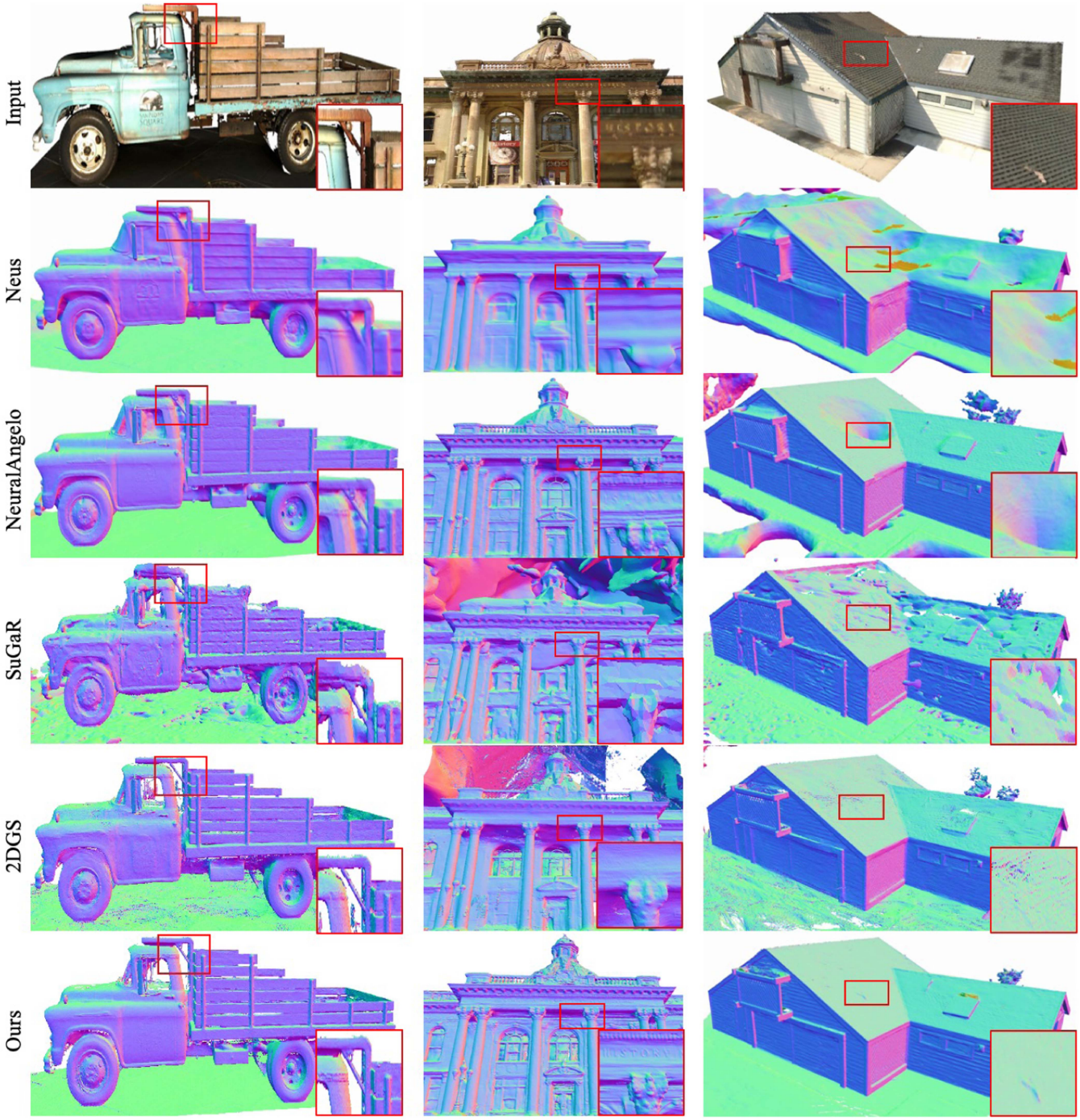

Fig. 8. Qualitative comparison on Tanks and Temples dataset. We visualize surface quality using a normal map generated from the reconstructed mesh. PGSR outperforms other baseline approaches in capturing scene details, whereas baseline methods exhibit missing or noisy surfaces.

$$\boldsymbol{H}_{rn} = \boldsymbol{K}_n \left(\boldsymbol{R}_{rn} - \frac{\boldsymbol{T}_{rn} \boldsymbol{n}_r^T}{d_r} \right) \boldsymbol{K}_r^{-1}, \tag{8}$$

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

$$\boldsymbol{L}_{mvgeom} = \frac{1}{V} \sum_{\boldsymbol{p}_r \in W} w(\boldsymbol{p}_r) \phi(\boldsymbol{p}_r), \tag{9}$$

$$w(\boldsymbol{p}_r) = \begin{cases} 1/\exp\left(\phi(\boldsymbol{p}_r)\right), & \text{if } \phi(\boldsymbol{p}_r) < 1 \\ 0, & \text{if } \phi\left(\boldsymbol{p}_r\right) >= 1 \end{cases}, \tag{10}$$

where $\phi(\boldsymbol{p}_r) = \| \boldsymbol{p}_r - \boldsymbol{H}_{nr}\boldsymbol{H}_{rn}\boldsymbol{p}_r \|$ is the forward and backward projection error of $\boldsymbol{p}_r$. When $\phi(\boldsymbol{p}_r)$ 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 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. $w(\boldsymbol{p}_r)$ is a weight of geometric occlusion estimation, and the larger the projection error, the smaller the weight. During training, the gradient of the weight will be detached.

Multi-View Photometric Consistency: Drawing inspiration from multi-view Stereo (MVS methods) [29], [34], [60], we employ photometric multi-view consistency constraints based on plane patches. We map a 7×7 pixel patch $\boldsymbol{P}_r$ centered at $\boldsymbol{p}_r$ to the neighboring frame patch $\boldsymbol{P}_n$ using the homography matrix $\boldsymbol{H}_{rn}$. Focusing on geometric details, we convert color images into grayscale. Multi-view photometric regularization requires that $\boldsymbol{P}_r$ and $\boldsymbol{P}_n$ should be as consistent as possible. We use the normalized cross correlation (NCC) [61] of patches in the reference frame and the neighboring frame to measure the photometric consistency:

$$\boldsymbol{L}_{mvrgb} = \frac{1}{V}\sum_{\boldsymbol{p}_r \in W} w(\boldsymbol{p}_r)\left(1 - NCC\left(\boldsymbol{I}_r\left(\boldsymbol{p}_r\right), \boldsymbol{I}_n\left(\boldsymbol{H}_{rn}\boldsymbol{p}_r\right)\right)\right), \tag{11}$$

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

$$\boldsymbol{L}_{geom} = \lambda_2 \boldsymbol{L}_{svgeom} + \lambda_3 \boldsymbol{L}_{mvrgb} + \lambda_4 \boldsymbol{L}_{mvgeom}. \tag{12}$$

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:

$$\boldsymbol{I}_i^a = \exp(a_i)\boldsymbol{I}_i^r + b_i, \tag{13}$$

where $\boldsymbol{I}_i^r$ is the rendered image and $\boldsymbol{I}_i^a$ is the exposure-adjusted image. We employ the following image loss:

$$\boldsymbol{L}_{rgb} = (1-\lambda)\boldsymbol{L}_1\left(\tilde{\boldsymbol{I}} - \boldsymbol{I}_i\right) + \lambda \boldsymbol{L}_{SSIM}(\boldsymbol{I}_i^r - \boldsymbol{I}_i). \tag{14}$$

$$\tilde{\boldsymbol{I}} = \begin{cases} \boldsymbol{I}_i^a, & \text{if } \boldsymbol{L}_{SSIM}(\boldsymbol{I}_i^r - \boldsymbol{I}_i) < 0.5 \\ \boldsymbol{I}_i^r, & \text{if } \boldsymbol{L}_{SSIM}(\boldsymbol{I}_i^r - \boldsymbol{I}_i) >= 0.5 \end{cases} \tag{15}$$

where $\boldsymbol{I}_i$ 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, $\boldsymbol{I}_i^r$ is required to be globally consistent and maintain structural similarity with the ground truth image, while $\boldsymbol{I}_i^a$ can adjust the brightness of images to match the ground truth image perfectly.

D. Training

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

$$\boldsymbol{L} = \boldsymbol{L}_{rgb} + \lambda_1 \boldsymbol{L}_s + \boldsymbol{L}_{geom}. \tag{16}$$

We set $\lambda_1 = 100$. For the image reconstruction loss, we set $\lambda = 0.2$. For the geometric loss, we set $\lambda_2 = 0.015$, $\lambda_3 = 0.15$, and $\lambda_4 = 0.03$.

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 [15] for evaluating novel view synthesis performance. The large and complex scenes of the TnT [24] 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) [63]. For assessing surface quality, we employed the F1 score and chamfer distance.

Implementation Details: Our training strategy and hyperparameters are generally consistent with 3DGS [20]. The training iterations for all scenes are set to 30,000. We adopt the densification strategy of AbsGS [64]. The initial value of the exposure coefficient is 0, and the learning rate is 0.001. We begin by rendering the depth for each training view, followed by utilizing the TSDF Fusion algorithm [65] to generate the corresponding TSDF field. Subsequently, we extract the mesh [66] from the TSDF field. We only utilize the exposure compensation on the Tanks and Temples dataset. All experiments in this paper are conducted on an Nvidia RTX 4090 GPU.

A. Real-Time Rendering

For the validation of rendering quality, we follow the 3DGS method and conduct validation on the Mip-NeRF360 dataset [15]. We compare with current state-of-the-art methods for pure novel view synthesis as well as similar reconstruction methods to ours, including NeRF [11], Deep Blending [67], INGP [17], Mip-NeRF360 [15], NeuS [19], 3DGS [20], SuGaR [21], 2DGS [55], and GOF [56]. As shown in Table I and Fig. 5, compared to the current state-of-the-art methods,

TABLE I
QUANTITATIVE RESULTS OF RENDERING QUALITY FOR NOVEL VIEW SYNTHESIS ON MIP-NERF360 DATASET

		Indoor scenes			Outdoor scenes			Average on all scenes		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF-based	NeRF [11]	26.84	0.790	0.370	21.46	0.458	0.515	24.15	0.624	0.443
	Deep Blending [67]	26.40	0.844	0.261	21.54	0.524	0.364	23.97	0.684	0.313
	INGP [17]	29.15	0.880	0.216	22.90	0.566	0.371	26.03	0.723	0.294
	M-NeRF360 [15]	31.72	0.917	0.180	24.47	0.691	0.283	28.10	0.804	0.232
	Neus [19]	25.10	0.789	0.319	21.93	0.629	0.600	23.74	0.720	0.439
GS-based	3DGS [20]	30.99	0.926	0.199	24.24	0.705	0.283	27.24	0.803	0.246
	SuGaR [21]	29.44	0.911	0.216	22.76	0.631	0.349	26.10	0.771	0.283
	2DGS [55]	30.39	0.923	0.183	24.33	0.709	0.284	27.03	0.804	0.239
	GOF [56]	30.80	0.928	0.167	24.76	0.742	0.225	27.78	0.835	0.196
	PGSR	30.36	0.934	0.147	24.76	0.752	0.203	27.25	0.833	0.178

"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.

TABLE II
QUANTITATIVE RESULTS OF CHAMFER DISTANCE(MM)↓ ON DTU DATASET [23]

	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	Mean	Time
VolSDF [51]	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 [19]	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 [18]	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 [21]	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 [55]	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 [56]	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	0.36	0.57	0.38	0.33	0.78	0.58	0.50	1.08	0.63	0.59	0.46	0.54	0.30	0.38	0.34	0.52	0.5h

PGSR achieves the highest reconstruction accuracy and is over 100 times faster than the SDF method based on NeRF.

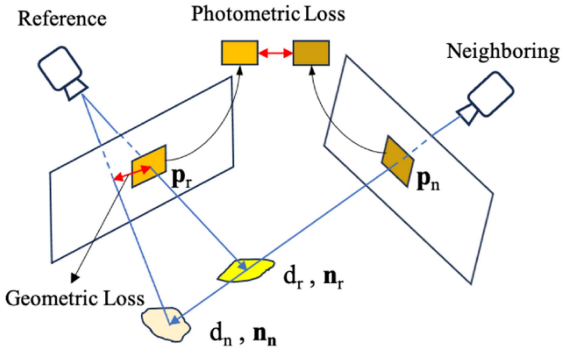

Fig. 9. Multi-view photometric and geometric loss.

TABLE III
QUANTITATIVE RESULTS OF F1 SCORE↑ FOR RECONSTRUCTION ON TANKS AND TEMPLES DATASET

	NeuS	Geo-Neus	Neurlangelo	SuGaR	2D GS	GOF	PGSR
Barn	0.29	0.33	0.70	0.14	0.45	0.51	0.66
Caterpillar	0.29	0.26	0.36	0.16	0.24	0.41	0.44
Courthouse	0.17	0.12	0.28	0.08	0.13	0.28	0.20
Ignatius	0.83	0.72	0.89	0.33	0.50	0.68	0.81
Meetingroom	0.24	0.20	0.32	0.15	0.18	0.28	0.33
Truck	0.45	0.45	0.48	0.26	0.43	0.58	0.66
Mean	0.38	0.35	0.50	0.19	0.32	0.46	0.52
Time	>24h	>24h	>128h	2h	0.57h	2h	0.75h

PGSR achieves the best reconstruction accuracy and very fast training speed.

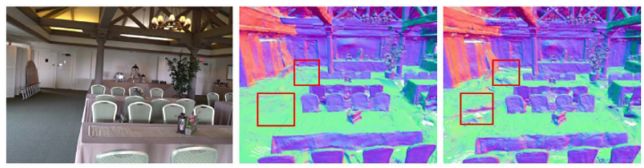

Fig. 10. The qualitative comparison of our unbiased depth method with the previous depth method [22], [68] is depicted in the normal map. Our overall geometric structure appears smoother and more precise.

our approach not only provides excellent surface reconstruction quality but also achieves outstanding novel view synthesis.

B. Reconstruction

We compared our method, PGSR, with current state-of-the-art neural surface reconstruction methods including NeuS [19], Geo-NeuS [60], VolSDF [51], and NeuralAngelo [18]. We also compared it with recently emerged reconstruction methods based on 3DGS, such as SuGaR [21], 2DGS [55], and GOF [56]. All results are summarized in Figs. 5, 7, 8, Tables II and III.

The DTU dataset: Our method achieves the highest reconstruction accuracy with relatively fast training speed. 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 precise, especially in flat regions. Table IV also demonstrates that our depth rendering method achieves higher reconstruction and rendering accuracy.

TABLE IV
ABLATION STUDY ON THE TNT DATASET

Model setting	F1-Score↑	PSNR↑
w/o Single-View	0.49	27.02
w/o Multi-View	0.32	27.30
w/o Multi-View Geometric	0.49	27.07
w/o Multi-View Photometric	0.39	26.83
w/o Geometric Occlusion Estimation	0.28	21.70
w/o Our unbiased depth	0.38	26.47
w/o Exposure Compensation	0.49	25.33
Full model	0.52	26.73

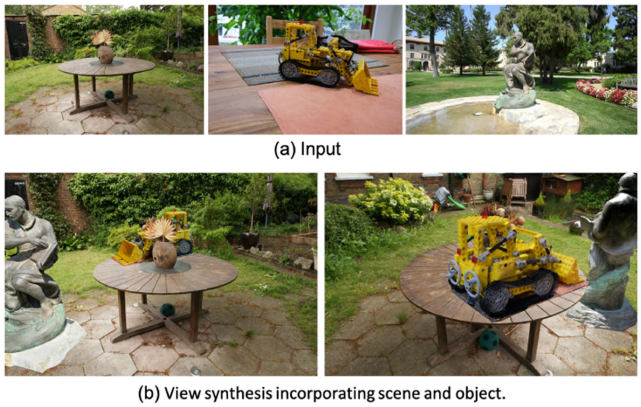

Fig. 11. Virtual Reality Application. (a) Original materials, including garden scene, excavator, and Ignatius. (b) A Virtual Reality effect showcase synthesized from these original materials.

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 constrains the consistency of geometry between multiple views, improving overall reconstruction accuracy. Both multi-view photometric and geometric consistency contribute to improving reconstruction accuracy. From Table IV, it is evident that multi-view regularization is crucial for reconstruction accuracy. However, without incorporating potential occlusion estimation, the multi-view regularization term will have a negative effect, leading to poor surface reconstruction and rendering accuracy.

The ablation results also reflect another issue: geometric constraints slightly degrade rendering quality. We speculate that this is due to an incomplete image rendering model, which forces the system to strike a balance between image and geometry losses. Further exploration may be needed to achieve synchronized improvements in geometry and novel view synthesis.

Exposure Compensation: As shown in Table IV, 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 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. First, 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. Second, 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 [69] would help further enhance the 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 show that our method achieves the highest geometric reconstruction accuracy and competitive rendering quality compared to state-of-the-art methods.

REFERENCES

[1] N. Deng et al., "Fov-NeRF: Foveated neural radiance fields for virtual reality," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 11, pp. 3854–3864, Nov. 2022.
[2] W. Ye, H. Li, T. Zhang, X. Zhou, and H. Bao, and G. Zhang, "SuperPlane: 3D plane detection and description from a single image," in *Proc. IEEE Virtual Reality 3D User Interfaces*, 2021, pp. 207–215.
[3] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3D using 2D diffusion," 2022, *arXiv:2209.14988*.
[4] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3D content creation," 2023, *arXiv:2309.16653*.
[5] P. Gao et al., "Lumina-T2X: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers," 2024, *arXiv:2405.05945*.
[6] Y. Chen et al., "Artist-created mesh generation with autoregressive transformers," 2024, *arXiv:2406.10163*.
[7] H. Li, W. Ye, G. Zhang, S. Zhang, and H. Bao, "Saliency guided subdivision for single-view mesh reconstruction," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 1098–1107.
[8] X. Liu, W. Ye, C. Tian, Z. Cui, H. Bao, and G. Zhang, "Coxgraph: Multi-robot collaborative, globally consistent, online dense reconstruction system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8722–8728.
[9] W. Ye et al., "PVO: Panoptic visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 9579–9589.
[10] W. Ye et al., "DeFlowSLAM: Self-supervised scene motion decomposition for dynamic dense SLAM," 2022, *arXiv:2207.08794*.

[11] B. Mildenhall et al., "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[12] W. Ye et al., "IntrinsicNeRF: Learning intrinsic neural radiance fields for editable novel view synthesis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 339–351.

[13] C. Huang et al., "NeRF-Det: Incorporating semantic cues and perspective-aware depth supervision for indoor multi-view 3D detection," 2024, *arXiv:2402.14464*.

[14] Y. Ming, W. Ye, and A. Calway, "iDF-SLAM: End-to-end RGB-D SLAM with neural implicit mapping and deep feature tracking," 2022, *arXiv:2209.07919*.

[15] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. Srinivasan, "Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5855–5864.

[16] J. T. Barron, B. Mildenhall, D. Verbin, P. PratulSrinivasan, and P. Hedman, "Zip-NeRF: Anti-aliased grid-based neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19697–19705.

[17] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–15, 2022.

[18] Z. Li et al., "Neuralangelo: High-fidelity neural surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8456–8465.

[19] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," 2021, *arXiv:2106.10689*.

[20] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 1–14, 2023.

[21] A. Guédon and V. Lepetit, "SuGaR: Surface-aligned gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering," 2023, *arXiv:2311.12775*.

[22] Y. Jiang et al., "Gaussianshader: 3D Gaussian splatting with shading functions for reflective surfaces," 2023, *arXiv:2311.17977*.

[23] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, "Large scale multi-view stereopsis evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 406–413.

[24] A. Knapitsch, J. Park, Q. Yi Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.

[25] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M Montiel, and J. D. Tardós, ""ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[26] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang, "RNIN-VIO: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2021, pp. 275–283.

[27] D. Chen et al., "VIP-SLAM: An efficient tightly-coupled RGB-D visual inertial planar SLAM," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 5615–5621.

[28] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Proc. Asian Comput. Vis. Conf.*, Springer, Berlin Heidelberg, 2012, pp. 257–270.

[29] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.

[30] C. Wu, "Towards linear-time incremental structure from motion," in *Proc. Int. Conf. 3D Vis.*, 2013, pp. 127–134.

[31] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.

[32] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418–433, Mar. 2005.

[33] N. Kiriakos Kutulakos and S. M. Seitz, "A theory of shape by space carving," *Int. J. Comput. Vis.*, vol. 38, pp. 199–218, 2000.

[34] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," in *Proc. 10th Eur. Conf. Comput. Vis.*, Marseille, France, Springer, 2008, pp. 766–779.

[35] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 873–881.

[36] J. Lutz Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 501–518.

[37] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, 2009, Art. no. 24.

[38] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction," in *Effective Computational Geometry for Curves and Surfaces*. Berlin, Germany: Springer, 2006, pp. 231–276.

[39] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.*, 2006, pp. 61–70.

[40] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[41] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12716–12725.

[42] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "PatchmatchNet: Learned multi-view patchmatch stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14194–14203.

[43] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2463–2471.

[44] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3D object reconstruction," in *Proc. Conf. Artif. Intell.*, 2018, pp. 7114–7121.

[45] C. B. Choy, D. Xu, J. Y. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 628–644.

[46] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2Vox: Context-aware 3D reconstruction from single and multi-view images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2690–2698.

[47] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 55–71.

[48] L. M. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4460–4470.

[49] J. J. Park, P. Florence, J. Straub, R. A. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[50] M. Niemeyer, L. M. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3501–3512.

[51] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 4805–4815.

[52] Q. Xu et al., "Point-NeRF: Point-based neural radiance fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5438–5448.

[53] H. Li, X. Yang, H. Zhai, Y. Liu, H. Bao, and G. Zhang, "Vox-surf: Voxel-based implicit surface representation," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 3, pp. 1743–1755, Mar. 2022.

[54] L. Liu, J. Gu, K. Z. Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 15651–15663.

[55] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D gaussian splatting for geometrically accurate radiance fields," 2024, *arXiv:2403.17888*.

[56] Z. Yu, T. Sattler, and A. Geiger, "Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes," 2024, *arXiv:2404.10772*.

[57] H. Chen, C. Li, and G. Hee Lee, "NeuSG: Neural implicit surface reconstruction with 3D gaussian splatting guidance," 2023, *arXiv:2312.00846*.

[58] X. Long et al., "Adaptive surface normal constraint for geometric estimation from monocular images," 2024, *arXiv:2402.05869*.

[59] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "GeoNet: Geometric neural network for joint depth and surface normal estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 283–291.

[60] Q. Fu, Q. Xu, Y. Soon Ong, and W. Tao, "Geo-NeuS: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 3403–3416.

[61] J.-C. Yoo and T. HeeHan, "Fast normalized cross-correlation," *Circuits, Syst. Signal Process.*, vol. 28, pp. 819–843, 2009.

[62] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian splatting SLAM," 2023, *arXiv:2312.06741*.

[63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[64] Z. Ye, W. Li, S. Liu, P. Qiao, and Y. Dou, "AbsGS: Recovering fine details for 3D gaussian splatting," 2024, *arXiv:2404.10484*.
[65] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.
[66] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Seminal Graph.: Pioneering Efforts Shaped Field*, vol. 1, pp. 347–353, 1998.
[67] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, "Deep blending for free-viewpoint image-based rendering," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–15, 2018.
[68] K. Cheng et al., "GaussianPro: 3D Gaussian splatting with progressive propagation," 2024, *arXiv:2402.14650*.
[69] T. Lu et al., "Scaffold-GS: Structured 3D Gaussians for view-adaptive rendering," 2023, *arXiv:2312.00109*.

Danpeng Chen is working toward the DEng degree with the State Key Lab of CAD&CG, Zhejiang University, advised by Prof. Hujun Bao and Prof. Guofeng Zhang. He also worked with Tetras.ai. His research interests include SLAM, 3D reconstruction, and their applications in virtual augmented reality.

Hai Li received the BS degree in computer science and technology from Harbin Engineering University, in 2016, and the PhD degree from Zhejiang University, in 2023. His research interests include 3D reconstruction, SLAM and augmented Reality.

Weicai Ye received the BS degree in software engineering from the University of Electronic Science and Technology of China, in 2018 and the PhD degree from Zhejiang University, in 2024. His research interest is 3D vision foundation model, and embodied AI, especially correspondence, 3D/4D reconstruction, rendering and generation.

Yifan Wang is currently working toward the bachelor's degree with Xi'an Jiaotong University. He is an intern with the Shanghai Artificial Intelligence Laboratory. His research interests include 3D reconstruction and 4D reconstruction.

Weijian Xie (Graduate Student Member, IEEE) received the master's degree in computer science from Zhejiang University, in 2017 and the DEng degree from Zhejiang University, in 2024. He is also working as senior algorithm scientist with SenseTime Research. His research interests include SLAM, 3D reconstruction, and augmented reality.

Shangjin Zhai received the master's degree from the State Key Lab of CAD&CG, Zhejiang University, where he specialized in the research area of Simultaneous Localization and Mapping (SLAM). He is currently working as a computer vision researcher with SenseTime Research.

Nan Wang received the master's degree from the State Key Lab of CAD&CG, Zhejiang University, in 2015, advised by Prof. Guofeng Zhang. He is currently working as research director with SenseTime. Before that, he was a senior R&D with Baidu Inc.. His research interests include SLAM, 3D reconstruction, and augmented reality.

Haoming Liu received the master and PhD degrees in computer science from Zhejiang University, in 2012 and 2017, respectively. He is currently a research director with SenseTime Research. His research interests include structure from motion, SLAM, augmented reality, and 3D AIGC.

Hujun Bao (Member, IEEE) is currently a professor with Computer Science Department, Zhejiang University, and the former director with the State Key Laboratory of Computer Aided Design and Computer Graphics. He leads the Mixed Reality Group, State Key Laboratory of Computer Aided Design and Computer Graphics, to make a wide range of research on 3D reconstruction and modeling, real-time rendering and virtual reality, real-time 3D fusion, and augmented reality. Some of these algorithms have been successfully integrated into the mixed reality system SenseMARS. His research interests include computer graphics, computer vision, and mixed reality.

Guofeng Zhang (Member, IEEE) received the BS. and PhD degrees in computer science and technology from Zhejiang University, in 2003 and 2009, respectively. He is currently a professor with Zhejiang University. His research interests include SLAM, 3D reconstruction, and augmented reality. He received the National Excellent Doctoral Dissertation Award, the Excellent Doctoral Dissertation Award of the China Computer Federation, and the ISMAR 2020 Best Paper Award.