PointSLAM++: Robust Dense Neural Gaussian Point Cloud-based SLAM

Xu Wang[1*], Boyao Han[1*], Xiaojun Chen[1], Ying Liu[2†], Ruihui Li[1†]

[1]College of Computer Science and Electronic Engineering, Hunan University, China
[2]College of Information Science and Engineering, Hunan Normal University, China
{xuwang0303, hby_22, chenxiaojun}@hnu.edu.cn, liu_ying@hunnu.edu.cn, liruihui@hnu.edu.cn

Abstract

Real-time 3D reconstruction is crucial for robotics and augmented reality, yet current simultaneous localization and mapping(SLAM) approaches often struggle to maintain structural consistency and robust pose estimation in the presence of depth noise. This work introduces PointSLAM++, a novel RGB-D SLAM system that leverages a hierarchically constrained neural Gaussian representation to preserve structural relationships while generating Gaussian primitives for scene mapping. It also employs progressive pose optimization to mitigate depth sensor noise, significantly enhancing localization accuracy. Furthermore, it utilizes a dynamic neural representation graph that adjusts the distribution of Gaussian nodes based on local geometric complexity, enabling the map to adapt to intricate scene details in real time. This combination yields high-precision 3D mapping and photorealistic scene rendering. Experimental results show PointSLAM++ outperforms existing 3DGS-based SLAM methods in reconstruction accuracy and rendering quality, demonstrating its advantages for large-scale AR and robotics.

Introduction

Visual SLAM uses monocular, stereo, or RGB-D images to reconstruct 3D scenes and estimate camera trajectories. Its widespread adoption in robotics, VR, and AR has intensified demands for higher-fidelity rendering and robust motion estimation (Zubizarreta, Aguinaga, and Montiel 2020). Traditional RGB-D SLAM (feature- or voxel-based) ensures stable tracking but yields coarse details (Du et al. 2011; Keller et al. 2013; Newcombe et al. 2011), while neural implicit methods (e.g., iMAP (Sucar et al. 2021), NICE-SLAM (Zhu et al. 2022a)) achieve dense reconstructions at high computational cost and limited scale. This efficiency–quality tradeoff has spurred new SLAM representations (Bylow et al. 2013; Canelhas, Stoyanov, and Lilienthal 2013; Dai et al. 2017; Prisacariu et al. 2017; Whelan et al. 2013). Recently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) emerged as a hybrid paradigm, using anisotropic Gaussians and differentiable splatting to combine point-cloud efficiency with neural-rendering fidelity (Zhu et al. 2024b). Systems like

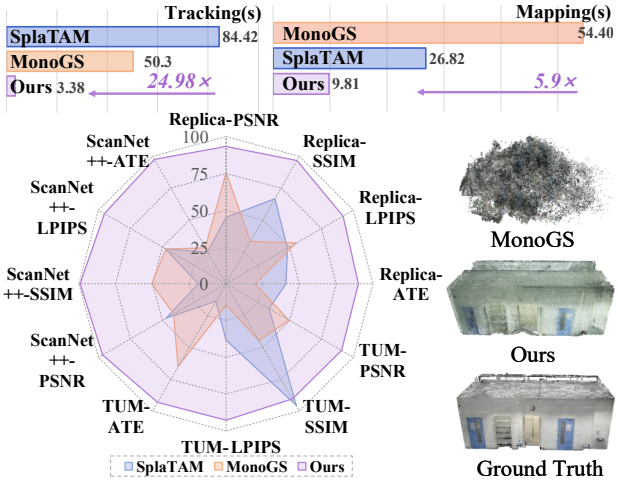

Figure 1: PointSLAM++ outperforms state-of-the-art methods from CVPR 2024 in photorealistic 3D reconstruction and camera tracking, excelling in key metrics. In complex environments where MonoGS fails, it maintains accurate localization and high-quality 3D mapping.

MonoGS (Matsuki et al. 2024) demonstrate real-time photorealistic reconstruction and joint pose–geometry optimization, though challenges remain with sensor noise, dynamic scenes, and large-scale environments (Huang et al. 2024; Keetha et al. 2024; Matsuki et al. 2024; Yan et al. 2024; Hu et al. 2024).

Current 3DGS-SLAM methods face three major challenges. First, reliance on precise depth information: an overdependence on high-quality depth maps causes the accuracy of tracking and reconstruction to fall in the presence of depth noise, missing data, or occlusions (Chung et al. 2023; Deng et al. 2024). Second, Gaussian sphere overfitting: existing techniques fit a Gaussian sphere to every training view, ignoring local scene structure, which introduces significant redundancy and limits scalability in complex, large-scale environments (Zhang et al. 2023). Third, weak view dependence: view-dependent effects are baked into a single Gaussian parameter, yielding poor interpolation capabilities and low robustness to extensive viewpoint and il-

*These authors contributed equally.
†Corresponding author.

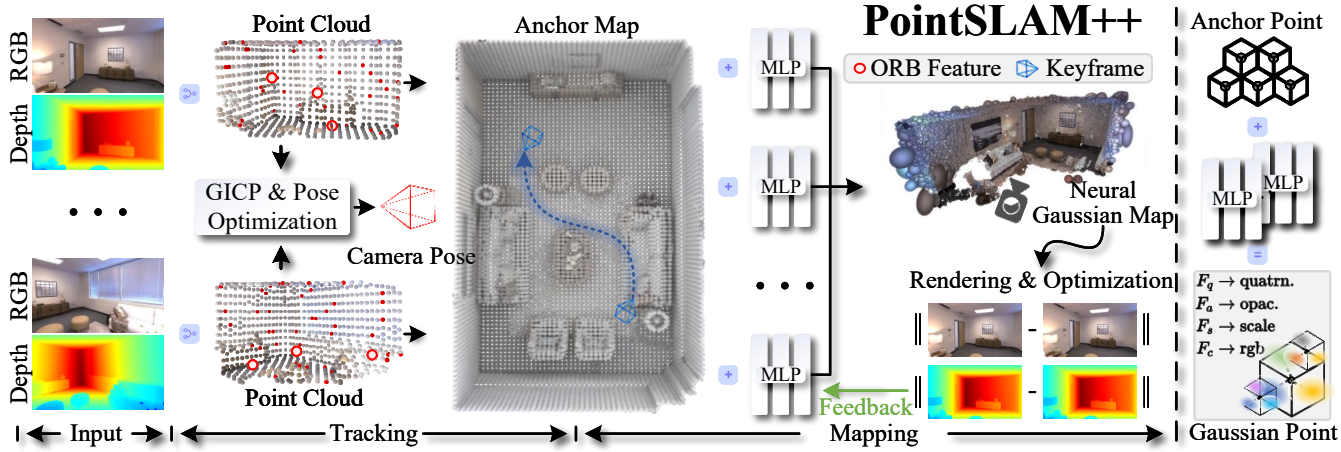

Figure 2: SLAM System Overview. The system's input is a sequence of RGB-D frames. We generate a point cloud by downsampling and reprojecting the current depth image, and we estimate the current pose using GICP and ORB features. During tracking, we create anchor points from the point cloud and utilize a neural network to predict the Gaussian distribution in the scene, rendering the scene using a specialized Gaussian rasterizer. We continuously optimize the multi-layer perceptron (MLP) during the mapping process using the RGB-D information. Right: We use anchor points and features as input to the MLP to predict the various attributes of the Gaussians.

lumination changes. Addressing these issues is essential for achieving scalable, high-fidelity 3DGS-SLAM.

This paper introduces PointSLAM++, a real-time 3D reconstruction system using RGB-D cameras. Using a learnable neural Gaussian representation with hierarchical constraints and progressive pose optimization, it enables precise mapping and realistic rendering of scenes.

To eliminate reliance on precise depth information, we propose a progressive optimization scheme to achieve accurate pose estimation: we apply the iterative closest point (ICP) algorithm for rigid registration of depth data and integrate ORB-feature-based visual constraints for joint refinement. Within a multi-scale pyramid, we simultaneously minimize the depth point-cloud residuals and the ORB-based visual reprojection errors (Campos et al. 2021). However, under challenging conditions—such as high-speed motion, low-texture regions, or dynamic interference—tracking can still be lost. To recover the camera pose quickly when conventional tracking fails, we introduce a relocalization mechanism based on global feature matching.

To meet the demands of real-time incremental mapping and high-precision reconstruction, stable ORB feature point clouds from tracking are fused with depth sensor data for map construction. The point cloud is then converted into neural Gaussian "anchors" endowed with position, scale, and related parameters, and their distributions are constrained by the scene's geometric structure. A two-tier anchor mechanism includes primary anchors for global structure and secondary anchors for fine detail, enabling incremental updates and detail enrichment while preserving robust pose estimation and geometric accuracy. Finally, camera-view vectors are embedded into each anchor's appearance features to compensate for viewpoint-dependent lighting and reflectance effects, thereby enhancing reconstruction quality under complex illumination and varying viewpoints.

In summary, our contributions are as follows:

- We present PointSLAM++, a real-time RGB-D SLAM system that unifies geometric and visual pose refinement with a hierarchical Neural-Gaussian representation for precise mapping and photorealistic rendering.
- We develop a progressive multi-scale pose optimization that jointly minimizes ICP depth residuals and ORB reprojection errors, and we add a global-feature relocalization module to recover from fast motion or low-texture tracking failures.
- We propose a two-tier neural-Gaussian anchor framework that fuses ORB feature clouds with depth data and embeds camera-view vectors to adaptively enhance detail and compensate for view-dependent lighting.

Related Work

Classical RGBD SLAM

Traditional RGB-D dense SLAM evolves through three stages: KinectFusion (Newcombe et al. 2011) achieved real-time TSDF reconstruction via ICP (Gelfand et al. 2003); ElasticFusion (Whelan et al. 2016) enhanced large-scale topological consistency through graph-based non-rigid deformation; and BundleFusion (Dai et al. 2017) improved robustness via hierarchical optimization and global loop closure. However, voxel and point-based semantic SLAM remain computationally heavy and inconsistent. SDF-based methods, while accurate, suffer from memory and scalability limits due to uniform grids (Wang, Skorokhodov, and Wonka 2022). Moreover, post-processing like CRF segmentation causes label misalignment and blurred boundaries from decoupled optimization.

Figure 3: GICP Mismatches in TUM Scenes. Even with high-precision depth data, real-world noise can lead to misalignments in GICP-based pose estimation, as illustrated here in the TUM dataset.

NeRF-based RGBD SLAM

The strong scene representation ability of Neural Radiance Fields (Mildenhall et al. 2020) (NeRF) has driven their use in 3D reconstruction and localization. NeRF-based SLAM builds continuous implicit functions via MLPs that map 3D coordinates to radiance and density (Wang, Wang, and Agapito 2023; Johari, Carta, and Fleuret 2023). This overcomes the resolution limits of voxel and point-based models, achieving sub-voxel accuracy in static scenes (e.g., iMAP (Sucar et al. 2021), NICE-SLAM (Zhu et al. 2022a)). Yet, challenges remain in optimization efficiency, dynamic scene disentanglement, and scalability, hindering robustness in large, real-world environments (Jiang et al. 2023).

Gaussian-based RGBD SLAM

3D Gaussian Splatting (Kerbl et al. 2023) has recently reshaped RGB-D SLAM. Approaches such as Gaussian-Splatting SLAM (Matsuki et al. 2024; Liso et al. 2024) exploit analytical Jacobians and geometric regularization for photorealistic reconstruction, surpassing NeRF in accuracy. GS-ICP SLAM (Ha, Yeon, and Yu 2024) further integrates Gaussian covariance into a G-ICP framework, achieving sub-millimeter alignment at twice the speed of conventional ICP. Remaining challenges include depth noise affecting covariance initialization, the need for heavy regularization, and limited dynamic scene modeling due to offline segmentation or simplified motion assumptions (Zhu et al. 2024a; Mao et al. 2024). Incremental optimization also causes misalignment in large-scale mapping. Addressing these issues requires noise-robust regularization and scalable dynamic modeling for practical SLAM deployment.

Method

An overview of our reconstruction pipeline is shown in Fig 2. In Section 3.1, we first introduce the progressive pose optimization(PPO) framework . Then, in Section 3.2, we describe the neural Gaussian representation and its density-adaptive optimization strategy. Finally, in Section 3.3, we provide a detailed description of the entire online reconstruction process based on neural Gaussian representation and progressive pose optimization.

Progressive Pose Optimization

In the camera pose localization problem, the primary objective is to accurately estimate camera trajectories within a global coordinate system. To achieve highly robust pose localization, we propose a progressive optimization framework that systematically refines pose estimates through cascaded stages, integrating point-cloud registration with feature-based refinement to mitigate common sources of error in real-world scenarios.

We begin with Generalized Iterative Closest Point (GICP) as the front-end odometry for coarse pose estimation. Given the local point cloud $\mathcal{P} = \{p_i \mid i = 1, 2, \ldots, N\}$ from the previous frame and $\mathcal{Q} = \{q_i \mid i = 1, 2, \ldots, N\}$ from the current frame, the optimal transformation $\mathbf{T} \in SE(3)$ minimizes the registration error:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \sum_{j=1}^{N} \sum_{i \in C(j)} d(\mathbf{T} \cdot q_j, p_i)^2 \tag{1}$$

where $C(j)$ denotes the correspondence set for point q_j, and $d(\cdot)$ is the distance metric. GICP iteratively optimizes correspondences and the transformation to yield an initial pose estimate.

Although accurate depth improves GICP, real-world noise often causes mismatches (Fig. 3). To mitigate this, we exploit the robustness of ORB features with more reliable depths. Extracted via multi-scale pyramids and fused with depth data, these features form local point clouds registered to the global map through point-to-plane constraints, initialized by the coarse GICP pose $\mathbf{T}$ for faster convergence and higher accuracy.

Building on this foundation, our framework further refines the pose by minimizing reprojection errors through bundle adjustment, which jointly optimizes camera poses and 3D landmarks. This progressive strategy—unique in its seamless fusion of odometry, feature-driven registration, and optimization—employs GICP+ORB for initialization, anchors the first frame to resolve scale ambiguity, parameterizes poses on the $SE(3)$ manifold, and utilizes a robust Levenberg-Marquardt solver with Huber loss and depth priors for stable convergence. The core objective unifies these elements:

$$\min_{\xi} \sum_{k=1}^{K} \rho\left(e_k^T \Sigma_k^{-1} e_k\right) \tag{2}$$

where: - $\xi \in \mathfrak{se}(3)$ parameterizes the pose $T = \exp(\xi^\wedge) \in SE(3)$, - K is the number of observed features, - $e_k = x_k - \pi(KTX_k) \in \mathbb{R}^2$ denotes the reprojection error, with x_k as observed pixels, π as the projection function, K as intrinsics, and X_k as 3D points, - $\Sigma_k^{-1} \in \mathbb{R}^{2\times 2}$ is the information matrix, - ρ is a robust kernel for outlier rejection.

Pose Recovery and Optimization. In challenging conditions such as high-speed motion, blur, sparse textures, or dynamic scenes, tracking may falter. Our recovery mechanism innovatively combines feature matching with established techniques to restore poses reliably. An initial correspondence set M is formed via descriptor matching be-

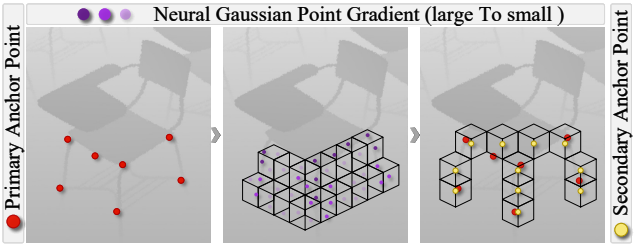

Figure 4: **Hierarchical Anchor Point Optimization.** Primary anchor points (red) are derived from ORB features and remain stable throughout SLAM, while secondary anchor points (yellow) are introduced or culled based on voxel-wise gradient checks (purple) of neural Gaussian points.

tween current features and map points. The Perspective-n-Point (PnP) solver, embedded in a RANSAC loop, then computes a robust initial pose by minimizing reprojection errors as per Equation (2), incorporating the robust kernel ρ to handle mismatches.

Subsequent nonlinear optimization refines this estimate using the same Lie algebra formulation and objective from Equation (2). This layered architecture—coarse GICP, ORB-augmented registration, bundle adjustment, and PnP+RANSAC recovery—sets our method apart by offering adaptive robustness that surpasses the isolated application of these components in existing pose tracking systems. Relocalization succeeds with high confidence when inlier rates exceed a predefined threshold, bolstering overall system resilience in complex environments.

Neural-Gaussian Representation

High-quality 3D reconstruction relies on accurate map initialization. While COLMAP-generated sparse point clouds serve as effective priors, they are not suitable for SLAM applications. To address this, we propose leveraging stable ORB feature point clouds extracted during SLAM tracking, supplemented by depth sensor data, for efficient and lightweight initialization.

At startup, the mapping thread seeds the global map with the first frame's ORB feature point cloud. It then incrementally incorporates feature points from each subsequent keyframe, continuously updating and expanding the map. To achieve high-fidelity scene reconstruction, we convert traditional ORB point clouds into neural point clouds with enhanced representational power:

$$N_v = \{(p_v, f_v, l_v, O_v)\} \tag{3}$$

Each neural Gaussian point consists of the center position $p_v \in \mathbb{R}^3$ of a voxel v, local context features $f_v \in \mathbb{R}^{32}$, scaling factors $l_v \in \mathbb{R}^3$, and k learnable offsets $O_v \in \mathbb{R}^{k\times 3}$. Similar to the Scaffold-GS method (Lu et al. 2024), we refer to these points as "anchor points".

Each anchor point generates K neural Gaussian models through a Multi-Layer Perceptron (MLP). Each neural Gaussian model is defined as:

$$G = \{(\mu, \alpha, q, s, c)\} \tag{4}$$

where $\mu \in \mathbb{R}^3$ is the 3D position, $\alpha \in \mathbb{R}$ is the opacity, $q \in \mathbb{R}^4$ is a quaternion controlling the covariance, $s \in \mathbb{R}^3$ is a scaling vector, and $c \in \mathbb{R}^3$ is the color.

Specifically, the relationship between the 3D position of a neural Gaussian model and its learnable offset is given by:

$$\{\mu_0, \ldots, \mu_{k-1}\} = x_v + \{O_0, \ldots, O_{k-1}\} \cdot l_v \tag{5}$$

where $\{O_0, O_1, \ldots, O_{k-1}\} \in \mathbb{R}^{k\times 3}$ represents a set of learnable offsets and l_v is the scaling factor associated with the anchor point.

The set of attributes, A, for the k neural Gaussians generated by each anchor point is computed by an MLP network F_a, as follows:

$$\{A_0, \ldots, A_{k-1}\} = F_a(\hat{f}_v, \delta_v c, \vec{d}_v c) \tag{6}$$

where A represents any attribute of the neural Gaussian (opacity α, color c, quaternion q, or scaling s). $\{A_0, \ldots, A_{k-1}\}$ denotes the set of that specific attribute for the k neural Gaussians generated by a single anchor point. Subsequently, rendering is performed using a Gaussian renderer. F_a represents the MLP network used to compute the specific attribute A. $\delta_v c$ and $\vec{d}_v c$ represent the relative viewing distance and direction between the camera and the anchor point. The specific calculation details will be provided in the supplementary materials.

Hierarchical Anchor Point Optimization. Unlike point clouds initialized using SfM reconstruction, the map maintained by SLAM algorithms is a dynamic, incremental representation. The Mapping and Tracking threads share the same map before the Tracking thread terminates. Therefore, extensive modifications to anchor points can directly impact camera pose estimation.

To address this, we introduce hierarchical anchor points. Anchor points generated from ORB features are designated as primary anchor points, which cannot undergo splitting or deletion and are optimized using a small learning rate. As ORB features have limited modeling capabilities in texture-scarce regions, we introduce secondary anchor points generated from depth sensor data and smaller voxels. Both primary and secondary anchors are collectively denoted as N_v.

To determine where secondary anchor points are needed, we construct voxels of size ϵ_g to spatially quantize neural Gaussian points. For each voxel, we calculate the average gradient ∇g over N training iterations. If $\nabla g > \tau_g$ and no anchor point exists in the voxel, a new secondary anchor point is deployed using depth information at the voxel center. If a secondary anchor point already exists, its opacity is halved until it falls below the threshold and is culled.

View-Dependent Environment Compensation. During camera motion, scene appearance varies due to viewpoint changes. These variations result from ambient lighting changes affecting camera exposure and material properties like specular reflections causing view-dependent radiance. These factors require neural Gaussians at the same location to render different pixel values from different viewpoints. However, traditional Gaussian models with their isotropic assumption cannot adequately handle these view-dependent appearance changes, even with spherical harmonics.

To address this limitation, we encode the camera viewing direction as a unit vector $\mathbf{v} \in \mathbb{R}^3$ and embed it together with the anchor appearance feature $\hat{\mathbf{f}}_a$ into the contextual features:

$$\mathbf{f}'_a = \mathrm{Embed}(\hat{\mathbf{f}}_a,\ \mathbf{v}) \tag{7}$$

where $\hat{\mathbf{f}}_a$ is the anchor appearance feature (i.e. the RGB color), $\mathbf{v}$ is the normalized view-direction vector, and $\mathrm{Embed}(\cdot)$ is implemented by an MLP. The output $\mathbf{f}'_a$ is the view-direction-enhanced contextual feature, which serves as the initial form of the contextual feature $\mathbf{f}_v$ and is then continuously updated during training. Hence, we denote this initial contextual feature by $\mathbf{f}_a$. This approach explicitly integrates view direction with appearance features and overcomes traditional methods' dynamic-viewpoint limitations.

SLAM System

We have developed a real-time SLAM system based on a neural Gaussian representation and a progressive pose optimization framework. Assume we have a map represented by a set of 3D Gaussians, constructed from previous camera frames 1 to t. For a new RGB-D frame $t+1$, our SLAM system performs the following steps:

- **Camera Tracking:** We estimate the camera pose by minimizing the reprojection error and depth error between the current frame and the global map, utilizing the image and depth information from the current RGB-D frame $t+1$. A coarse estimate is first obtained using GICP, followed by optimization using point cloud constraints based on ORB features and the reprojection error.
- **Gaussian Densification:** When the current frame $t+1$ is determined to be a keyframe, we add new anchor points to the map based on the ORB feature points and their corresponding depth information from that frame. This enhances the map's ability to represent fine details.
- **Map Update:** If the current frame $t+1$ is selected as a keyframe, we update the parameters of all neural Gaussians in the map by minimizing the RGB and depth errors on the current keyframe, as well as on historical keyframes that have overlap with the current keyframe. If the current frame is not a keyframe, a subset of historical keyframes is randomly selected for optimization to maintain continuous map refinement and consistency.

Keyframe Selection. We use a dynamic keyframe selection strategy, similar to GS-ICP SLAM (Ha, Yeon, and Yu 2024), based on geometric consistency. A new keyframe is selected if the average point cloud distance between the current frame and the map exceeds a threshold, or if too few points fall below that threshold. To mitigate error accumulation, new Gaussian points are only added to non-overlapping regions of the new keyframe.

Map Update. During anchor point optimization, we employ multiple losses—color, SSIM (based on 3D Gaussian splatting), and geometric—to ensure color fidelity, structural consistency, and geometric coherence in rendered images. A scale regularization term further guides Gaussian ellipsoids toward isotropic spheres, reducing anisotropy and balancing tracking accuracy with rendering realism.

$$\begin{aligned}\mathcal{L}_{mapping} = \lambda_{I_1}\mathcal{L}_1\left(I, I_{gt}\right) + \lambda_{I_2}\mathcal{L}_{D\text{-}SSIM}\left(I, I_{gt}\right) \\ + \lambda_D\mathcal{L}_1\left(D, D_{gt}\right)\end{aligned} \tag{8}$$

Experiments

Experimental Setup

Implementation Details. Our SLAM system is implemented on a desktop computer equipped with an Intel Xeon Silver 4314 CPU and an Nvidia RTX 3090 GPU. The mapping and tracking components are implemented in Python and built upon the Pytorch framework.

Datasets. We evaluate our method on three datasets. Replica (Straub et al. 2019) provides high-precision synthetic RGB-D images suitable for basic validation. TUM-RGBD dataset (Sturm et al. 2012), with precise camera poses from motion capture systems, is standard for SLAM tracking accuracy evaluation despite its low image quality and noisy depth data. ScanNet++ (Yeshwanth et al. 2023) is a large-scale dataset with both high and ordinary quality indoor scene data, featuring larger camera pose intervals and depth map errors, allowing us to test system robustness under fast camera movement or sparse texture conditions.

Baselines. We compare our method with state-of-the-art RGB-D SLAM approaches, including NICE-SLAM (Zhu et al. 2022b), Point-SLAM (Sandström et al. 2023), and concurrent Gaussian SLAM methods such as SplaTAM (Keetha et al. 2024), MonoGS (Matsuki et al. 2024), Photo-SLAM (Huang et al. 2024), and GS-ICP SLAM (Ha, Yeon, and Yu 2024). To ensure the credibility of the results, we reproduce the experiments using the official code provided by each method. For the ScanNet++ dataset (Yeshwanth et al. 2023), due to its sparse viewpoint distribution, we double the scene (i.e., the map in SLAM) update frequency for all methods to ensure a fair comparison.

Quality Of Reconstructed Map

Table 1 shows that PointSLAM++ significantly outperforms other methods in novel view rendering on RGB-D scenes across TUM-RGBD (Sturm et al. 2012) and Replica (Straub et al. 2019) datasets.

On Replica (Straub et al. 2019), with its high-quality depth data and simpler scenes, PointSLAM++ achieves optimal results in all sequences.Among the comparative approaches, the state-of-the-art GS-ICP SLAM ranks second in rendering precision. PointSLAM++ successfully surpasses GS-ICP SLAM by leveraging neural Gaussian techniques. The implementation of a unique anchor point optimization technique skillfully addresses appearance variations, enabling PointSLAM++ to excel even under ideal conditions and outperform GS-ICP SLAM in rendering quality.

The TUM-RGBD dataset (Sturm et al. 2012) presents greater challenges with small objects, motion blur, and incomplete depth maps. Despite these difficulties, PointSLAM++ maintains the highest rendering accuracy

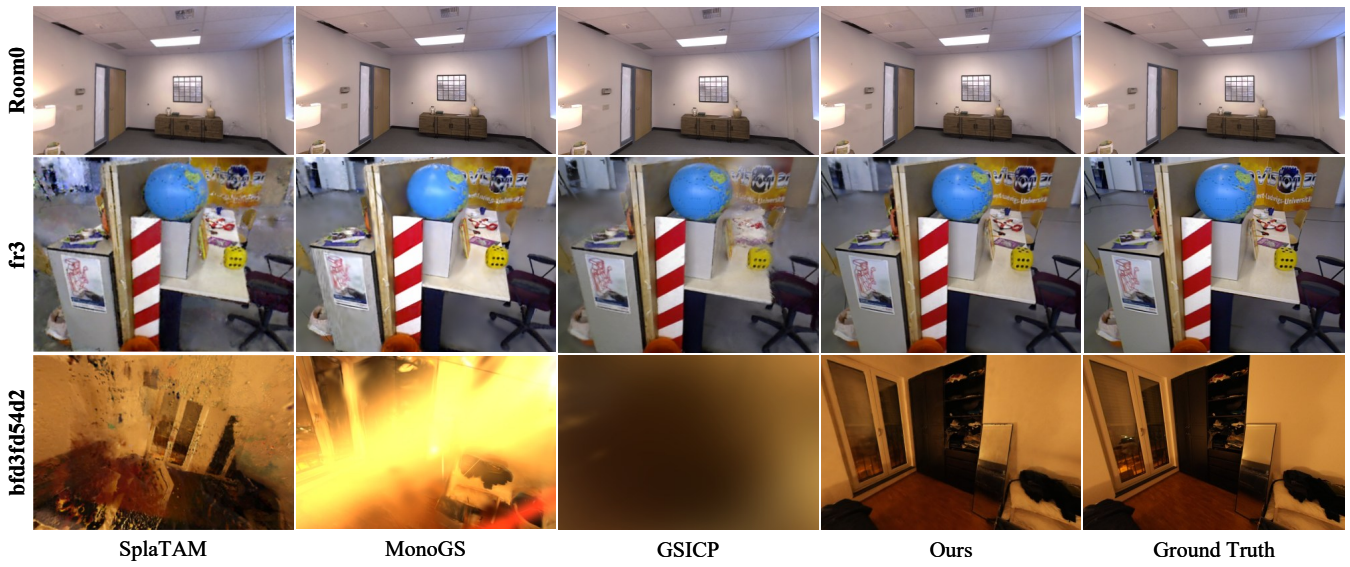

Figure 5: **Comparison of Rendering Results.** We present scenes from three datasets: Room0 (Replica), fr3 (TUM-RGBD), and bfd3fd54d2 (ScanNet++). Our method achieves rendering results that closely match the ground truth, demonstrating high accuracy. Additional results are available in the supplementary materials.

Method	Replica				ScanNet++			
	PSNR↑	SSIM↑	LPIPS↓	ATE RMSE↓	PSNR↑	SSIM↑	LPIPS↓	ATE RMSE↓
NICE-SLAM (Zhu et al. 2022a)	24.42	0.809	0.233	1.95	-	-	-	-
Point-SLAM (Sandström et al. 2023)	5.17	0.975	0.124	0.52	-	-	-	-
SplaTAM (Keetha et al. 2024)	34.11	0.970	0.100	0.36	14.32	0.482	0.599	287.36
MonoGS (Matsuki et al. 2024)	37.5	0.960	0.070	0.58	12.90	0.648	0.603	101.19
Photo-SLAM (Huang et al. 2024)	34.96	0.942	0.059	0.61	×	×	×	×
GS-ICP SLAM (Ha, Yeon, and Yu 2024)	38.83	0.975	0.041	**0.16**	14.94	0.776	0.446	111.37
Ours	**39.46**	**0.979**	**0.027**	0.19	**26.51**	**0.905**	**0.148**	**6.73**

Table 1: Quantitative evaluation of ours compared to existing methods for RGB-D cameras on Replica (Straub et al. 2019) and ScanNet++ (Yeshwanth et al. 2023) datasets, where '–' denotes systems failing to provide valid results and '×' indicates systems unable to complete tracking or reconstructio. More quantitative results can be found in the supply materials.

across all sequences. GS-ICP SLAM performs poorly here due to its over-reliance on depth maps. PointSLAM++ uses depth primarily to enhance localization and geometry rather than letting depth noise directly affect rendering, significantly improving performance.

On ScanNet++ (Yeshwanth et al. 2023), PointSLAM++ achieves a decisive victory where methods like GS-ICP SLAM fail to complete camera pose localization. PointSLAM++ enables high-quality map reconstruction and novel view rendering while maintaining stable localization, demonstrating its exceptional capability with challenging real-world scan data.

Camera Tracking Accuracy

Table 1 shows our method's superior tracking accuracy on the Replica dataset, reducing trajectory errors by over 50% compared to alternatives. By leveraging high-precision depth images and 3D information from G-ICP, we outperform methods relying on 2D spatial errors. Our performance is comparable to GS-ICP SLAM, as high-precision depth enables accurate camera pose predictions with GICP.

Table 3 shows our results on the TUM-RGBD dataset, our method achieves exceptional tracking accuracy through progressive pose optimization. By incorporating ORB features, we mitigate depth noise interference, enhancing localization stability. In contrast, GS-ICP SLAM performs poorly with noisy depth data, as its GICP algorithm struggles to deliver reliable results.

The evaluation results for the ScanNet++ dataset are listed in Table 1, our algorithm demonstrates tracking stability in scenarios where other methods fail. It addresses the dataset's challenges: large camera pose intervals, sparse textures, and depth interference. While GICP methods fail under these conditions and approaches relying on 2D spatial errors falter with substantial pose intervals, our method remains robust, showing its adaptability to complex environments. Additional results are in supply materials.

Method	Metric	fr1/desk	fr2/xyz	fr3/office	Avg.
NICE-SLAM	PSNR↑	13.83	17.87	12.89	14.86
	SSIM↑	0.569	0.718	0.554	0.614
	LPIPS↓	0.482	0.344	0.498	0.441
Point-SLAM	PSNR↑	13.87	17.56	18.43	16.62
	SSIM↑	0.627	0.708	0.754	0.696
	LPIPS↓	0.546	0.585	0.448	0.526
SplaTAM	PSNR↑	22.00	24.50	21.90	22.80
	SSIM↑	0.860	**0.950**	0.880	**0.897**
	LPIPS↓	0.230	0.100	0.200	0.177
MonoGS	PSNR↑	23.69	24.68	22.83	23.7329
	SSIM↑	0.786	0.793	0.770	0.783
	LPIPS↓	0.245	0.225	0.277	0.249
Photo-SLAM	PSNR↑	20.87	22.10	22.74	21.90
	SSIM↑	0.743	0.765	0.780	0.763
	LPIPS↓	0.239	0.169	0.145	0.184
GS-ICP SLAM	PSNR↑	17.95	23.13	20.50	20.53
	SSIM↑	0.710	0.829	0.758	0.766
	LPIPS↓	0.296	0.141	0.232	0.223
Ours	PSNR↑	**25.05**	**27.11**	**26.33**	**26.16**
	SSIM↑	**0.876**	0.899	**0.881**	0.885
	LPIPS↓	**0.123**	**0.078**	**0.118**	**0.106**

Table 2: Rendering Performance on TUM-RGBD (Sturm et al. 2012).

Method	fr1/desk	fr2/xyz	fr3/office	Avg.
NICE-SLAM	4.26	31.73	3.87	13.29
Point-SLAM	4.34	1.31	3.48	3.04
SplaTAM	3.35	1.24	5.16	3.25
MonoGS	**1.50**	1.44	1.49	1.48
Photo-SLAM	2.60	0.35	**1.00**	1.32
GS-ICP SLAM	3.50	1.76	2.74	2.67
Ours	1.56	**0.33**	1.34	**1.08**

Table 3: Tracking Performance on TUM-RGBD (Sturm et al. 2012) (ATE RMSE [cm] ↓).

Efficiency And Memory Comparison

Table 4 compares PointSLAM++'s efficiency and memory with existing methods on the ScanNet++ (Yeshwanth et al. 2023). Compared to neural methods like SplaTAM and MonoGS, PointSLAM++ achieves high-quality reconstruction with significantly reduced processing time, due to our non-neural tracking optimization.

Ablation Study

We isolated two key modules from our progressive pose optimization framework and view-dependent environment compensation—and evaluated their impact through targeted experiments. Quantitative results are presented in Table 5.

Progressive Pose Optimization Framework. We compared the performance of PointSLAM++ with and without its progressive pose optimization (PPO). The results show a significant deterioration in tracking performance when relying solely on the GICP algorithm. This issue is pronounced under conditions with depth noise, where GICP alone leads to cumulative errors in geometrically ambiguous scenes. In contrast, the progressive approach effectively suppresses depth noise through its multi-stage adjustments. This method enhances the system's overall robustness and accuracy, proving beneficial in challenging scenarios involving rapid camera movement or dramatic changes in lighting.

Method	Mapping Time↓	Tracking Time↓	FPS↑	GPU Memory(GB)↓
SplaTAM	41m54s	117m49s	0.053	20.32
MonoGS	29m3s	17m29s	0.191	23.76
GS-ICP SLAM	**1m10s**	1m10s	2.84	**9.62**
Ours	8m13s	**1m00s**	**3.33**	10.11

Table 4: Comparison of Time Efficiency and Memory Consumption on the ScanNet++ (Yeshwanth et al. 2023).

Method	PSNR↑	SSIM↑	LPIPS↓	ATE RMSE↓
w/o PPO	21.47	0.784	0.201	17.33
w/o $NeuGS\&VDC$	24.06	0.812	0.168	2.32
w/o VDC	25.36	0.856	0.128	1.10
Ours	**26.16**	**0.885**	**0.106**	**1.08**

Table 5: Progressive Pose Optimization and Neural Gaussian Representation Ablation on TUM-RGBD (Sturm et al. 2012).

Neural Gaussian and View-Dependent Environment Compensation. We established two ablation variants: one without neural Gaussians (NeuGS) and view-dependent compensation (VDC), and a second without only the VDC. The results show that incorporating neural Gaussians improves reconstruction quality over the baseline, particularly enhancing detail representation and geometric consistency. While this intermediate version captures richer geometric and appearance information, the full PointSLAM++ model demonstrates superior rendering consistency across different viewpoints and greater adaptability to lighting conditions. This confirms that the view-dependent compensation mechanism is crucial for addressing rendering artifacts caused by significant viewpoint changes, with improvements being evident in scenes with dramatic lighting variations. Furthermore, we also observed that although the tracking components are identical in the variants, the unique optimization process of the neural Gaussians leads to a more accurate map. This enhanced map fidelity, in turn, contributes to more precise pose estimation.

Conclusion

We propose PointSLAM++, a real-time RGB-D reconstruction system based on neural Gaussian representation. By integrating progressive pose optimization and view-dependent compensation, it mitigates depth noise, captures fine geometry and appearance, and ensures cross-view rendering consistency. Experiments demonstrate superior reconstruction accuracy and robustness for robotics, mixed reality, and intelligent interaction. Despite its effectiveness, the method increases computational cost, motivating future work on improving efficiency for broader device deployment.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. U25A20421, No. 62202151, No. 62202152) and the National Key Research and Development Program of China (No. 2025YFB3003601).

References

Bylow, E.; Sturm, J.; Kerl, C.; Kahl, F.; and Cremers, D. 2013. Real-time camera tracking and 3D reconstruction using signed distance functions. In *Robotics: Science and systems (RSS) conference 2013*, volume 9. Robotics: Science and Systems.

Campos, C.; Elvira, R.; Rodríguez, J. J. G.; Montiel, J. M.; and Tardós, J. D. 2021. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6): 1874–1890.

Canelhas, D. R.; Stoyanov, T.; and Lilienthal, A. J. 2013. SDF tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3671–3676. IEEE.

Chung, C.-M.; Tseng, Y.-C.; Hsu, Y.-C.; Shi, X.-Q.; Hua, Y.-H.; Yeh, J.-F.; Chen, W.-C.; Chen, Y.-T.; and Hsu, W. H. 2023. Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 9400–9406. IEEE.

Dai, A.; Nießner, M.; Zollhöfer, M.; Izadi, S.; and Theobalt, C. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4): 1.

Deng, T.; Shen, G.; Qin, T.; Wang, J.; Zhao, W.; Wang, J.; Wang, D.; and Chen, W. 2024. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19657–19666.

Du, H.; Henry, P.; Ren, X.; Cheng, M.; Goldman, D. B.; Seitz, S. M.; and Fox, D. 2011. Interactive 3D modeling of indoor environments with a consumer depth camera. In *Proceedings of the 13th international conference on Ubiquitous computing*, 75–84.

Gelfand, N.; Ikemoto, L.; Rusinkiewicz, S.; and Levoy, M. 2003. Geometrically stable sampling for the ICP algorithm. In *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, 260–267. IEEE.

Ha, S.; Yeon, J.; and Yu, H. 2024. Rgbd gs-icp slam. In *European Conference on Computer Vision*, 180–197. Springer.

Hu, J.; Chen, X.; Feng, B.; Li, G.; Yang, L.; Bao, H.; Zhang, G.; and Cui, Z. 2024. Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In *European Conference on Computer Vision*, 93–112. Springer.

Huang, H.; Li, L.; Hui, C.; and Yeung, S.-K. 2024. Photo-SLAM: Real-time Simultaneous Localization and Photorealistic Mapping for Monocular, Stereo, and RGB-D Cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Jiang, Y.; Hedman, P.; Mildenhall, B.; Xu, D.; Barron, J. T.; Wang, Z.; and Xue, T. 2023. Alignerf: High-fidelity neural radiance fields via alignment-aware training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 46–55.

Johari, M. M.; Carta, C.; and Fleuret, F. 2023. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17408–17419.

Keetha, N.; Karhade, J.; Jatavallabhula, K. M.; Yang, G.; Scherer, S.; Ramanan, D.; and Luiten, J. 2024. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21357–21366.

Keller, M.; Lefloch, D.; Lambers, M.; Izadi, S.; Weyrich, T.; and Kolb, A. 2013. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, 1–8. IEEE.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Liso, L.; Sandström, E.; Yugay, V.; Van Gool, L.; and Oswald, M. R. 2024. Loopy-slam: Dense neural slam with loop closures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20363–20373.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20654–20664.

Mao, Y.; Yu, X.; Zhang, Z.; Wang, K.; Wang, Y.; Xiong, R.; and Liao, Y. 2024. Ngel-slam: Neural implicit representation-based global consistent low-latency slam system. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 6952–6958. IEEE.

Matsuki, H.; Murai, R.; Kelly, P. H.; and Davison, A. J. 2024. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18039–18048.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European Conference on Computer Vision (ECCV)*, 405–421.

Newcombe, R. A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A. J.; Kohi, P.; Shotton, J.; Hodges, S.; and Fitzgibbon, A. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, 127–136. Ieee.

Prisacariu, V. A.; Kähler, O.; Golodetz, S.; Sapienza, M.; Cavallari, T.; Torr, P. H. S.; and Murray, D. W. 2017. InfiniTAM v3: A Framework for Large-Scale 3D Reconstruction with Loop Closure. arXiv:1708.00783.

Sandström, E.; Li, Y.; Van Gool, L.; and R. Oswald, M. 2023. Point-SLAM: Dense Neural Point Cloud-based SLAM. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Straub, J.; Whelan, T.; Ma, L.; Chen, Y.; Wijmans, E.; Green, S.; Engel, J. J.; Mur-Artal, R.; Ren, C.; Verma, S.; Clarkson, A.; Yan, M.; Budge, B.; Yan, Y.; Pan, X.; Yon, J.; Zou, Y.; Leon, K.; Carter, N.; Briales, J.; Gillingham, T.; Mueggler, E.; Pesqueira, L.; Savva, M.; Batra, D.; Strasdat, H. M.; Nardi, R. D.; Goesele, M.; Lovegrove, S.; and Newcombe, R. 2019. The Replica Dataset: A Digital Replica of Indoor Spaces. *arXiv preprint arXiv:1906.05797*.

Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; and Cremers, D. 2012. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*.

Sucar, E.; Liu, S.; Ortiz, J.; and Davison, A. J. 2021. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6229–6238.

Wang, H.; Wang, J.; and Agapito, L. 2023. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13293–13302.

Wang, Y.; Skorokhodov, I.; and Wonka, P. 2022. Hf-neus: Improved surface reconstruction using high-frequency details. *Advances in Neural Information Processing Systems*, 35: 1966–1978.

Whelan, T.; Johannsson, H.; Kaess, M.; Leonard, J. J.; and McDonald, J. 2013. Robust real-time visual odometry for dense RGB-D mapping. In *2013 IEEE International Conference on Robotics and Automation*, 5724–5731. IEEE.

Whelan, T.; Salas-Moreno, R. F.; Glocker, B.; Davison, A. J.; and Leutenegger, S. 2016. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, 35(14): 1697–1716.

Yan, C.; Qu, D.; Xu, D.; Zhao, B.; Wang, Z.; Wang, D.; and Li, X. 2024. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19595–19604.

Yeshwanth, C.; Liu, Y.-C.; Nießner, M.; and Dai, A. 2023. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12–22.

Zhang, W.; Sun, T.; Wang, S.; Cheng, Q.; and Haala, N. 2023. Hi-slam: Monocular real-time dense mapping with hybrid implicit fields. *IEEE Robotics and Automation Letters*, 9(2): 1548–1555.

Zhu, L.; Li, Y.; Sandström, E.; Huang, S.; Schindler, K.; and Armeni, I. 2024a. Loopsplat: Loop closure by registering 3d gaussian splats. *arXiv preprint arXiv:2408.10154*.

Zhu, S.; Wang, G.; Blum, H.; Liu, J.; Song, L.; Pollefeys, M.; and Wang, H. 2024b. Sni-slam: Semantic neural implicit slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21167–21177.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022a. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12786–12796.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022b. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zubizarreta, J.; Aguinaga, I.; and Montiel, J. M. M. 2020. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4): 1363–1370.