Bridging the Gap Between Gaussian Splatting and SLAM: A Geometric-Gaussian Field-based Gaussian Splatting SLAM System

Jiasheng He[1], **Wenbin Zhu**[1*], **Xiaobin Liu**[1], **Jing Yuan**[1] and **Yuhang Wei**[1]
[1]College of Artificial Intelligence, Nankai University
2120230574@mail.nankai.edu.cn, wenbinzhu@nankai.edu.cn, liuxb@nankai.edu.cn, nkyuanjing@gmail.com, 2120240533@mail.nankai.edu.cn

Abstract

Recent works in Gaussian Splatting (GS) SLAM highlight the importance of geometric structure. However, existing methods often rely on either 2D or 3D Gaussian primitives, lacking the balance between geometry and appearance, thus failing to precisely model spatial structures. Furthermore, current frameworks either solely utilize SLAM to provide poses for GS or optimize poses and all Gaussian parameters indiscriminately, which neglect geometric consistency constraints essential for robust localization, creating a gap between GS and SLAM. In this paper, a tight-coupling SLAM system based on a Geometric-Gaussian Field (GGF) is proposed. First, GGF combines explicit spatial structures with implicit neural residuals to adaptively regulate the Gaussian morphologies, seamlessly transitioning between 2D and 3D Gaussian primitives, ensuring precise spatial modeling. Second, a Geometric Consistency-Guided Refinement (GCGR) strategy is introduced, which exploits depth and normal maps rendered by GGF to construct critical constraints for accurate pose estimation. Simultaneously, updated poses and spatial information are fed back to refine Gaussian morphology, establishing mutual enhancement and tight-coupling between GS and SLAM. Extensive experiments demonstrate that GGF-SLAM outperforms state-of-the-art methods in tracking accuracy, photorealistic rendering, and geometric reconstruction.

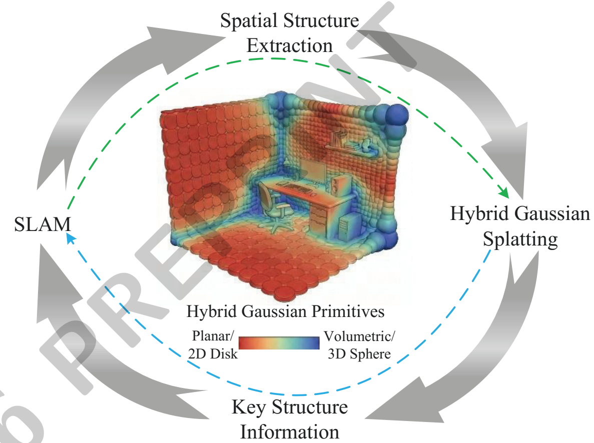


Figure 1: The tightly coupled framework of GGF-SLAM. The solid gray cycle demonstrates our proposed bidirectional mutual enhancement between SLAM and Hybrid Gaussian Splatting, effectively bridging the gap present in existing unidirectional or loosely coupled frameworks (dashed arrows) .

1 Introduction

Visual Simultaneous Localization and Mapping (SLAM) has achieved remarkable success in empowering autonomous systems to perceive and navigate unknown environments. Traditional methods [Mur-Artal *et al.*, 2015; Mur-Artal and Tardós, 2017; Campos *et al.*, 2021; Engel *et al.*, 2017; Klein and Murray, 2007; Davison *et al.*, 2007; Engel *et al.*, 2014; Forster *et al.*, 2014; Kaess, 2015; Pumarola *et al.*, 2017; Taguchi *et al.*, 2013; Hsiao *et al.*, 2017], which rely on sparse geometric features such as points, lines, and planes, excel at robust tracking and pose estimation. However, the resulting maps are inherently rudimentary, capturing only sparse structural skeletons that lack the dense textural information required for higher-level intelligent tasks. In this context, the emergence of 3D Gaussian Splatting (GS) [Kerbl *et al.*, 2023] has opened new possibilities for high-fidelity mapping in SLAM. By leveraging explicit yet differentiable primitives, 3DGS enables real-time photorealistic rendering, prompting a surge of research into integrating GS with SLAM systems [Huang *et al.*, 2024b; Keetha *et al.*, 2024; Matsuki *et al.*, 2024; Yan *et al.*, 2024; Li *et al.*, 2024] to achieve both accurate localization and precise scene reconstruction.

Despite these advancements, a deep fusion between GS and SLAM remains elusive, primarily due to the limitations of current Gaussian representations. Standard 3D Gaussian primitives prioritize photometric accuracy but lack the precise geometric structures required for reliable SLAM constraints. Conversely, 2D Gaussian primitives [Huang *et al.*, 2024a] incorporate geometric information but struggle with non-planar regions and exhibit limited visual expressiveness compared to their 3D counterparts. Even hybrid methods relying on hard

*Corresponding author.

pixel-wise classification introduce significant inconsistency, failing to provide the unified, high-quality representation necessary for tightly coupled localization and mapping.

Furthermore, current GS-SLAM frameworks are typically loosely coupled. Existing methods either unidirectionally utilize SLAM poses for mapping [Huang *et al.*, 2024b] (as indicated by the green arrow in Figure 1) or indiscriminately optimize all parameters based solely on photometric residuals [Keetha *et al.*, 2024] (as indicated by the blue arrow in Figure 1), ignoring the inherent complementarity of the two tasks. Actually, GS and SLAM can be further integrated: the key environmental information from GS can be leveraged to refine tracking, while the global priors from SLAM can be utilized to constrain Gaussian primitives. The absence of such mutual enhancement results in a gap between GS and SLAM.

To bridge this gap, a novel tight-coupling framework called GGF-SLAM is proposed, which constructs a bidirectional mutual enhancement loop. Central to GGF-SLAM is the design of the Geometric-Gaussian Field (GGF), which serves as a cohesive intermediary between SLAM and GS. GGF combines explicit structural priors with implicit neural residuals to adaptively regulate Gaussian morphologies, enabling a seamless and differentiable transition between 2D and 3D Gaussian primitives. This structure-aware representation allows us to address the two sub-problems of mutual enhancement: (1) SLAM enhancing GS: The refined poses and spatial information are fed back to GGF, using the long-term structural consistency of SLAM to guide the morphological evolution of Gaussians. This embeds accurate spatial structural information into Gaussian models, thereby simultaneously improving rendering fidelity and establishing a more precise 3D map. (2) GS enhancing SLAM: Leveraging the high-fidelity geometry maintained by GGF, we introduce a Geometric Consistency-Guided Refinement (GCGR) strategy, which extracts reliable depth and normal constraints from GGF to rectify camera drifts. Consequently, the tight coupling between high-fidelity mapping and robust localization can be established, as indicated by the gray arrows in Figure 1.

In summary, our main contributions are as follows:

- GGF-SLAM is proposed in this paper. To the best of our knowledge, this is the first tightly coupled system that bridges the gap between Gaussian Splatting and SLAM, which simultaneously leverages spatial structure and photometric information to establish a bidirectional mutual enhancement mechanism.
- The Geometric-Gaussian Field (GGF) is designed, which fuses explicit spatial structures with implicit neural residuals to adaptively regulate Gaussian morphologies, ensuring precise modeling of both planar and unstructured regions. Crucially, it incorporates long-term stable structural information provided by SLAM into Gaussian primitives, thereby significantly improving both 2D rendering fidelity and 3D mapping accuracy.
- A Geometric Consistency-Guided Refinement (GCGR) strategy is introduced, which utilizes GGF to extract key structural information for constructing the Gaussian map and designs a joint optimization objective, thereby achieving the effective enhancement of SLAM localization by Gaussian Splatting.
- We conduct extensive evaluations on widely used datasets, demonstrating that GGF-SLAM significantly outperforms state-of-the-art (SOTA) methods in tracking accuracy, photorealistic rendering, and geometric reconstruction.

2 Related Work

2.1 Conventional SLAM

Traditional SLAM systems are primarily categorized into sparse and dense methods. Sparse approaches, such as the ORB-SLAM series [Mur-Artal *et al.*, 2015; Campos *et al.*, 2021], rely on geometric features for robust tracking but yield sparse ”skeletons” lacking the dense texture required for scene understanding. Conversely, dense methods like KinectFusion [Newcombe *et al.*, 2011] and ElasticFusion [Whelan *et al.*, 2015] utilize explicit representations (e.g., TSDFs) to model geometry. However, their discrete, non-differentiable nature precludes gradient-based refinement, limiting their ability to achieve high-fidelity reconstruction.

2.2 NeRF-based SLAM

NeRF-based SLAM[Sucar *et al.*, 2021; Zhu *et al.*, 2022; Zhang *et al.*, 2023; Xin *et al.*, 2024; Wang *et al.*, 2023; Liso *et al.*, 2024] introduces implicit neural representations for continuous scene modeling. iMAP [Sucar *et al.*, 2021] employs a single MLP for joint tracking and mapping, while NICE-SLAM [Zhu *et al.*, 2022] and Co-SLAM [Wang *et al.*, 2023] incorporate hierarchical feature grids to enhance scene representation. Despite enabling differentiable optimization, these methods are bottlenecked by computationally intensive volumetric ray sampling. This inefficiency restricts rendering speeds and hinders high-frequency map updates.

2.3 Gaussian Splatting SLAM

3D Gaussian Splatting (3DGS) [Kerbl *et al.*, 2023] has revolutionized dense SLAM by enabling real-time, differentiable rendering, yet achieving a robust fusion remains challenging. Early approaches like Photo-SLAM [Huang *et al.*, 2024b] treat mapping as a unidirectional downstream task, failing to exploit dense map information for tracking refinement. Conversely, fully unified frameworks such as SplaTAM [Keetha *et al.*, 2024] indiscriminately optimize parameters based on photometric residuals, often overfitting appearance and producing geometric artifacts like ”blobs” that degrade tracking [Matsuki *et al.*, 2024]. Although recent methods attempt to enforce structure using 2D surfels [Huang *et al.*, 2024a] or point priors [Wen *et al.*, 2025], they typically compromise representational expressiveness or lack mechanisms for continuous bidirectional refinement, thus failing to achieve a tight coupling between GS and SLAM.

3 Preliminaries

The proposed system is constructed upon the explicit structured 3D Gaussian Splatting representation [Lu *et al.*, 2024] and utilizes a robust visual SLAM frontend [Campos *et al.*, 2021]. This section briefly outlines these foundational components.

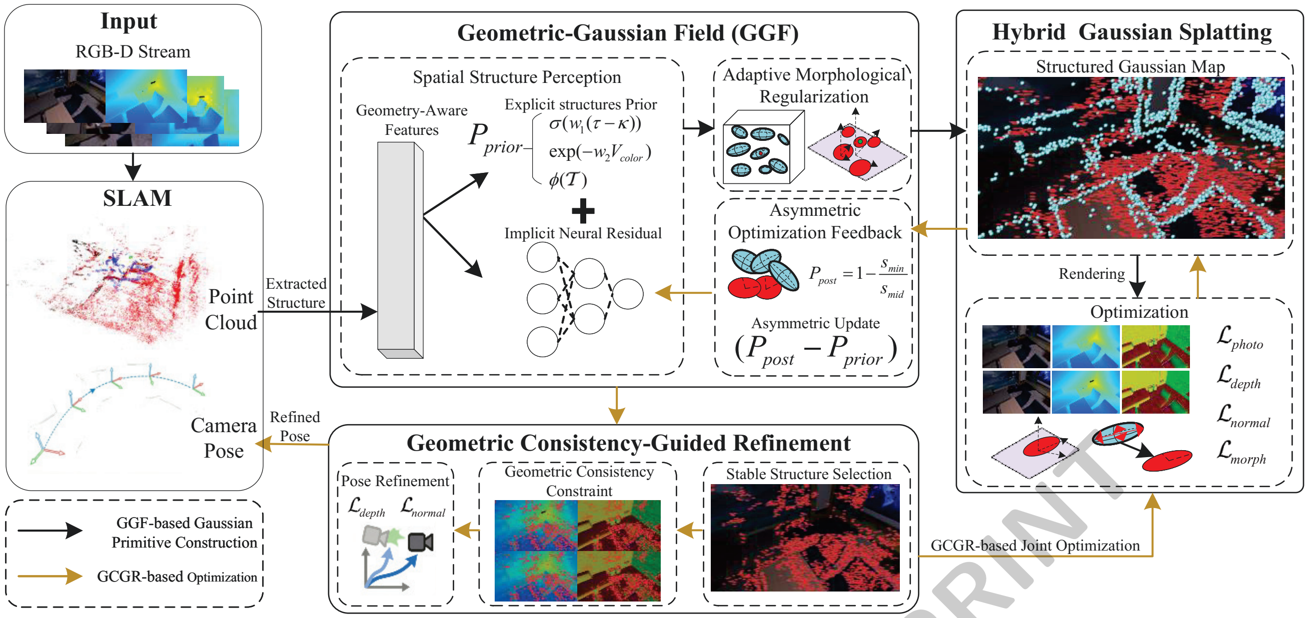


Figure 2: System framework of GGF-SLAM. The framework achieves tight coupling between SLAM and GS through the Geometric-Gaussian Field (GGF). The workflow comprises two streams: the construction stream (black arrows), which fuses explicit structural priors from SLAM with implicit neural residuals to generate hybrid Gaussians; and the optimization stream (golden arrows), which employs Geometric Consistency-Guided Refinement (GCGR) and asymmetric feedback to jointly refine camera poses and Gaussian attributes.

3.1 Structured 3D Gaussian Splatting

To efficiently represent scene geometry and appearance, we adopt the hierarchical Structured 3D Gaussians representation [Lu *et al.*, 2024]. Unlike the original 3DGS which optimizes discrete points directly, this representation generates and manages 3D Gaussian primitives based on a set of voxelized Anchor Points $\mathbf{p}_v$. Each anchor is associated with a feature vector $\mathbf{f}_v$, a scaling factor l_v, and a set of learnable offsets $\mathcal{O}$. Each anchor spawns k neural Gaussians, whose attributes (opacity, color, rotation, scale) are decoded from $\mathbf{f}_v$ via lightweight MLPs. Once the decoded attributes are obtained, the pixel color $C(\mathbf{u})$ is synthesized via α-blending by extending the differentiable tile-based rasterization pipeline:

$$C(\mathbf{u}) = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (1)$$

where $\mathbf{u}$ denotes the pixel coordinates, $\mathcal{N}$ is the set of sorted Gaussians contributing to the pixel, and α_i represents the final evaluated opacity.

3.2 Tracking and Mapping

Leveraging ORB-SLAM3 [Campos *et al.*, 2021], our frontend simultaneously estimates the camera state $\mathbf{T}_k = (\mathbf{R}_k, \mathbf{t}_k) \in SE(3)$ and reconstructs a point cloud $\mathcal{P}_{map} = \{\mathbf{P}_0, \ldots, \mathbf{P}_\eta\}$ via bundle adjustment. $\mathcal{P}_{map}$ serves as the geometric foundation of our framework: it not only facilitates the voxel-based initialization of the Gaussian anchors but also supplies the raw geometric attributes, such as local curvature and normal vectors, as spatial structural priors for the proposed GGF.

4 Methodology

4.1 System Overview

In this paper, a tight-coupling GS SLAM system called GGF-SLAM is proposed to bridge the critical gap between GS and SLAM. As illustrated in Figure 2, the GGF-SLAM framework comprises two main streams: GGF-based Gaussian primitives construction and GCGR-based optimization. In GGF-based Gaussian primitives construction, the SLAM module firstly processes incoming RGB-D frames to estimate initial camera poses $\mathbf{T}$ and generates a point cloud $\mathcal{P}_{map}$, from which we extract explicit structural attributes representing the long-term stable geometry captured by SLAM. Then, GGF fuses these explicit structural attributes with implicit neural residuals to adaptively regulate the morphology of Gaussian primitives. After that, the Adaptive Morphological Regularization (AMR) module completes the initialization of hybrid 3D and 2D Gaussian primitives. Finally, Gaussian splatting optimization is applied to obtain a precise structured Gaussian map. In GCGR-based optimization, the structured Gaussian map is first employed to refine the parameters of residual network within Asymmetric Optimization Feedback (AOF). Next, the key spatial structure information is extracted from GGF to construct the optimization function. Subsequently, within GCGR, a joint optimization objective combining photometric, geometric, and structural terms is formulated. By minimizing this objective, the system effectively rectifies both camera poses and Gaussian primitives. Consequently, the refined poses and updated Gaussian primitives are fed back into the SLAM and GS modules. Through these two distinct streams anchored by GGF, the system achieves mutual enhancement and tight coupling between GS and SLAM.

4.2 Geometric-Gaussian Field (GGF)

Unlike previous methods that rely on distinct 2D or 3D primitives, GGF serves as a continuous perception module. It actively infers the local geometric attributes of scene elements to enforce morphological constraints for Gaussian primitives.

Geometry-Aware Feature Space

To endow GGF with precise geometric perception, the raw geometric attributes extracted from the frontend are explicitly encoded into a high-dimensional feature. For each anchor point $\mathbf{p}$, we construct an input vector $\mathbf{F}_{in} \in \mathbb{R}^{41}$ defined as:

$$\mathbf{F}_{in} = [\gamma(\mathbf{p}), \mathbf{n}, \kappa, \mathcal{T}], \tag{2}$$

where $\gamma(\mathbf{p}) \in \mathbb{R}^{36}$ is the sinusoidal positional encoding capturing high-frequency spatial information; $\mathbf{n} \in \mathbb{R}^3$ is the normalized local surface normal; $\kappa \in \mathbb{R}^1$ is the local curvature scalar; and $\mathcal{T} \in \mathbb{R}^1$ represents observation confidence based on the number of tracked frames. Additionally, we calculate local color variance V_{color} as an independent criterion for photometric consistency. These terms provide the explicit geometric grounding necessary for inferring planarity, preventing the network from overfitting to photometric appearance.

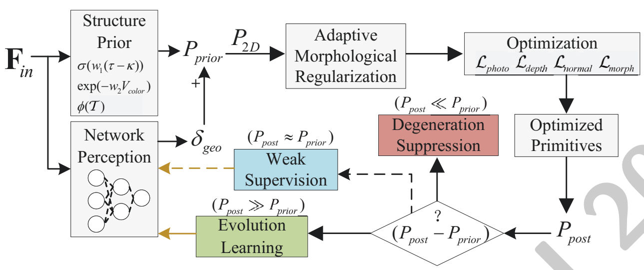


Figure 3: Workflow of GGF Inference and Asymmetric Optimization Feedback.

Spatial Structure Perception

To synergize the stability of explicit priors with the flexibility of data-driven learning, a hybrid inference mechanism is designed to dynamically estimate the planar probability P_{2D}. Firstly, a prior score P_{prior} is formulated based on the observation that valid planar regions statistically exhibit minimal curvature and high photometric consistency:

$$P_{prior} = \sigma(w_1(\tau - \kappa)) \cdot \exp(-w_2 V_{color}) \cdot \phi(\mathcal{T}), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function; τ denotes the curvature threshold; w_1, w_2 are weighting factors; and $\phi(\mathcal{T})$ represents a confidence based on tracked frame count. P_{prior} ensures the system maintains a stable geometric anchor in well-structured areas, preventing initialization collapse. However, explicit definitions may be insufficient for unstructured content. Therefore, we employ a lightweight network Φ_θ to perceive latent cues from the feature $\mathbf{F}_{in}$ and generate a residual correction term δ_{geo}:

$$\delta_{geo} = \Phi_\theta(\mathbf{F}_{in}). \tag{4}$$

Once δ_{geo} is acquired, it is integrated with P_{prior} to derive the final planar probability. To ensure mathematical consistency of fusion, P_{prior} is mapped into the latent space before integrating the residual. The final planar probability P_{2D} is derived as:

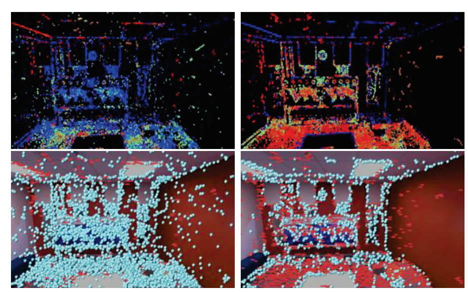
$$P_{2D} = \sigma\left(\sigma^{-1}(P_{prior}) + \delta_{geo}\right). \tag{5}$$

Instead of relying on static rules, this design strikes a robust balance between geometric constraints and flexible adaptation to diverse scene contexts, thereby continuously enhancing the system's perceptive capability across complex environments.

Adaptive Morphological Regularization

Driven by the inferred probability P_{2D}, a differentiable regularization scheme is implemented to dynamically modulate Gaussian primitives (as shown in Figure 4). For regions identified as planar ($P_{2D} > \tau_p$), a strict structural consistency is enforced by coupling orientation alignment with metric scaling. Specifically, the rotation quaternion $\mathbf{q}$ is constrained to align the Gaussian's shortest axis strictly with the surface normal $\mathbf{n}$, while simultaneously compressing the associated scale (s_{min}) to a minimal value ϵ. This allows for the adaptive transformation of 3D Gaussian primitives into 2D Gaussian primitives, resulting in a better fit for planar surfaces.

To prevent coverage gaps caused by regularization, we further employ a KNN-based Adaptive Scaling strategy. This dynamically adjusts tangential radii (s_{mid}, s_{max}) based on local neighbor density, ensuring seamless surface reconstruction. By strictly imposing these regularizations, GGF effectively prevents structural degradation (e.g., "blobs" or artifacts), maintaining high-fidelity reconstruction even under rigorous photometric optimization.



Without GGF **With GGF**

Figure 4: Impact of GGF on geometry reconstruction. Red and blue represent the distribution of 2D and 3D primitives, respectively.

Gaussian Optimization

To harmonize photorealism with structural fidelity, we optimize Gaussian parameters Θ via a joint objective that synergizes appearance, geometry, and morphology:

$$\mathcal{L}_{total} = \mathcal{L}_{photo} + \lambda_{geo}\mathcal{L}_{geo} + \lambda_{reg}\mathcal{L}_{morph}, \tag{6}$$

where $\mathcal{L}_{photo}$ ensures visual fidelity via $\mathcal{L}_1$ and D-SSIM constraints. Crucially, to resolve geometry-appearance ambiguity, $\mathcal{L}_{geo}$ is explicitly defined as the geometric loss comprising depth and normal constraints (formulated later in Eq. 11), which enforces dense alignment between rendered depth/normals and sensor observations. Complementing this, $\mathcal{L}_{morph}$ incorporates GGF inference to regularize planar primitives explicitly on orientation and scale:

$$\mathcal{L}_{morph} = \lambda_{dir} \underbrace{(1 - |\mathbf{r}_z^\top \mathbf{n}_{gt}|)}_{\text{Normal Alignment}} + \lambda_{scale} \underbrace{\max(0, s_z - \epsilon)}_{\text{Scale Constraint}}, \quad (7)$$

where $\mathbf{r}_z$ denotes the Gaussian's shortest axis (normal direction) and $\mathbf{n}_{gt}$ is the target geometric normal.

This optimization ensures the map converges to a structurally consistent state, providing a stable foundation for the subsequent feedback loop.

Asymmetric Optimization Feedback

To enable continuous and steady structural learning, an Asymmetric Optimization Feedback mechanism is introduced. Viewing GGF learning through a Bayesian lens, we treat the prior score of morphology as the explicit prior P_{prior}, while the geometric information obtained from the optimized Gaussian primitives serves as the optimized posterior P_{post}. The discrepancy between them is leveraged to selectively refine the residual perception module Φ_θ.

P_{post} is quantified via the primitive's morphological flatness. Defined by the ratio of its sorted scales $\mathbf{s} = [s_{min}, s_{mid}, s_{max}]$, the posterior probability is:

$$P_{post} = 1 - \frac{s_{min}}{s_{mid}}. \quad (8)$$

A high P_{post} signals that the primitive has successfully converged into a distinct planar structure.

While P_{post} reflects the optimization consensus, it carries the risk of overfitting. Therefore, we design an asymmetric supervision strategy based on the divergence between the prior and posterior ($P_{post} - P_{prior}$):

- **Evolution Learning** ($P_{post} \gg P_{prior}$)**:** If the posterior significantly exceeds the prior, it implies the optimization has recovered latent planar structures that rule-based priors missed. In this case, P_{post} is used as pseudo-ground truth to update the network, expanding its perceptive capacity.
- **Degeneration Suppression** ($P_{post} \ll P_{prior}$)**:** Conversely, if the posterior drops significantly below the prior, it indicates the primitive has degenerated into an unstructured "blob" ($s_{min} \approx s_{mid}$) to overfit appearance. In this case, P_{post} is identified as unreliable, and we strictly suppress the update, forcing the network to anchor to the stable P_{prior} to preserve geometric consistency.
- **Consistency Consolidation** ($P_{post} \approx P_{prior}$)**:** When the prior and posterior align, the refined P_{post} is utilized for supervision, but we attenuate the loss with a decay factor λ_{con}. This allows the network to gently absorb subtle geometric refinements without overwriting the established structural consensus.

As illustrated in Figure 3, this asymmetric mechanism enables GGF to strictly filter out geometric collapse while aggressively assimilating new structural cues, ensuring a symbiosis between stability and adaptability.

4.3 Geometric Consistency-Guided Refinement (GCGR)

In this section, GCGR is proposed to address the critical lack of deep synergy between Gaussian optimization and camera tracking in standard GS SLAM. In the GCGR module, the high-fidelity structure anchored by GGF is exploited to establish dense geometric constraints, thereby enhancing SLAM localization via Gaussian Splatting.

Differentiable Geometric Derivation

In GGF-SLAM, the rasterization pipeline is extended to generate dense geometric maps for constructing constraints. Crucially, to ensure precise pose optimization, stable structural regions are selected by GGF, parameterized by flattened 2D Gaussian primitives, to establish these dense constraints, thereby filtering out geometric ambiguity from unstructured blobs. Specifically, the expected depth $\hat{D}$ is first rendered via alpha-blending. Additionally, the normal map $\hat{N}$ is derived through differentiable back-projection. Pixels are lifted to 3D vertices $\mathcal{V}(\mathbf{u})$ using $\hat{D}$, and surface normals are computed as the normalized cross-product of tangent vectors derived from finite differences:

$$\begin{aligned} \hat{D}(\mathbf{u}) &= \sum_{i \in \mathcal{N}} d_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \\ \hat{N}(\mathbf{u}) &= \text{normalize}\left(\partial_u \mathcal{V}(\mathbf{u}) \times \partial_v \mathcal{V}(\mathbf{u})\right). \end{aligned} \quad (9)$$

This fully differentiable formulation establishes a direct gradient flow from geometric errors back to the camera pose $\mathbf{T}$, enabling structure-aware tracking.

Tight-Coupling Optimization

The camera pose $\mathbf{T} \in SE(3)$ is refined by minimizing a holistic objective that enforces geometric consistency across the scene. Specifically, the consistency is established through the alignment of both depth ($\mathcal{L}_{depth}$) and surface orientation ($\mathcal{L}_{normal}$):

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} (\lambda_{depth} \mathcal{L}_{depth} + \lambda_{normal} \mathcal{L}_{normal}). \quad (10)$$

The terms $\mathcal{L}_{depth}$ and $\mathcal{L}_{normal}$ are formulated as:

$$\begin{aligned} \mathcal{L}_{depth} &= \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \|\hat{D}(\mathbf{u}) - D_{gt}(\mathbf{u})\|_1, \\ \mathcal{L}_{normal} &= \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \left(1 - \hat{N}(\mathbf{u}) \cdot N_{gt}(\mathbf{u})\right), \end{aligned} \quad (11)$$

where Ω denotes the valid pixel set, and N_{gt} is the ground truth normal computed from sensor depth D_{gt}. We use the L_1 norm for depth and the cosine distance for normal alignment. Building upon the proposed GCGR-based optimization, GGF-SLAM leverages GGF to extract key structural information from the structured Gaussian primitives within the Gaussian Splatting module. By establishing geometric consistency constraints, this formulation enables the Gaussian representation to effectively enhance SLAM localization. In conjunction with GGF-based primitive construction, the proposed GGF-SLAM achieves tight coupling and bidirectional enhancement between GS and SLAM.

Datasets		Replica									TUM RGB-D			
Method	Metric	R0	R1	R2	Of0	Of1	Of2	Of3	Of4	Avg.	fr1/d	fr2/x	fr3/o	Avg.
MonoGS[Matsuki *et al.*, 2024]	PSNR↑	34.29	35.77	36.79	40.87	40.73	35.22	35.89	34.98	36.81	23.59	24.46	24.29	24.11
	SSIM↑	0.953	0.957	0.965	0.979	0.977	0.961	0.962	0.955	0.964	0.783	0.789	0.829	0.800
	LPIPS↓	0.071	0.078	0.074	0.048	0.052	0.074	0.061	0.092	0.069	0.244	0.227	0.223	0.231
Photo-SLAM[Huang *et al.*, 2024b]	PSNR↑	32.09	34.15	35.91	38.70	39.53	33.13	34.15	36.35	35.50	20.14	22.15	20.68	20.99
	SSIM↑	0.920	0.941	0.959	0.967	0.964	0.943	0.943	0.956	0.949	0.722	0.765	0.721	0.736
	LPIPS↓	0.069	0.055	0.041	0.048	0.045	0.075	0.064	0.053	0.056	0.258	0.169	0.211	0.213
RTG-SLAM[Peng *et al.*, 2024]	PSNR↑	28.49	31.27	32.96	37.32	36.12	31.14	31.19	33.81	32.79	13.62	17.08	18.70	16.47
	SSIM↑	0.834	0.902	0.927	0.957	0.943	0.923	0.918	0.937	0.918	0.501	0.573	0.648	0.574
	LPIPS↓	0.152	0.119	0.122	0.084	0.103	0.145	0.139	0.125	0.124	0.557	0.403	0.422	0.461
GS-SLAM[Yan *et al.*, 2024]	PSNR↑	31.56	32.86	32.59	38.70	41.17	32.36	32.03	32.92	34.27	-	-	-	-
	SSIM↑	0.968	0.973	0.971	0.986	**0.993**	0.978	0.970	0.968	0.975	-	-	-	-
	LPIPS↓	0.094	0.075	0.093	0.050	0.033	0.094	0.110	0.112	0.082	-	-	-	-
SplaTAM[Keetha *et al.*, 2024]	PSNR↑	32.54	33.58	35.03	38.00	38.85	31.71	29.74	31.40	33.85	21.02	23.39	19.81	21.41
	SSIM↑	0.938	0.936	0.952	0.963	0.955	0.928	0.902	0.914	0.936	0.753	0.806	0.731	0.764
	LPIPS↓	0.068	0.096	0.072	0.087	0.095	0.100	0.119	0.157	0.099	0.341	0.204	0.249	0.265
SGS-SLAM[Li *et al.*, 2024]	PSNR↑	32.48	33.50	35.11	38.22	38.91	31.86	30.05	31.53	33.96	-	-	-	-
	SSIM↑	**0.975**	0.968	**0.983**	0.983	0.982	0.966	0.952	0.946	0.969	-	-	-	-
	LPIPS↓	0.071	0.099	0.073	0.083	0.091	0.099	0.118	0.154	0.099	-	-	-	-
GS-ICP SLAM[Ha *et al.*, 2024]	PSNR↑	34.89	37.15	37.89	41.62	42.86	32.69	31.45	38.54	37.14	15.67	18.49	19.25	17.81
	SSIM↑	0.955	0.965	0.970	0.981	0.981	0.965	0.959	0.969	0.968	0.574	0.667	0.692	0.642
	LPIPS↓	0.048	0.045	0.047	0.027	0.031	0.057	0.057	0.045	0.045	0.444	0.308	0.329	0.361
SEGS-SLAM[Wen *et al.*, 2025]	PSNR↑	37.07	39.54	40.33	42.04	43.21	36.38	37.18	39.62	39.42	25.29	26.35	26.46	26.03
	SSIM↑	0.968	**0.977**	0.980	0.982	0.979	0.967	0.969	0.977	0.975	0.839	0.831	0.859	0.843
	LPIPS↓	0.023	0.016	**0.015**	**0.020**	**0.019**	0.035	0.026	0.018	0.021	**0.136**	0.081	0.105	0.107
Ours	PSNR↑	**38.17**	**40.34**	**40.96**	**43.64**	**44.35**	**37.86**	**38.25**	**40.50**	**40.51**	**26.56**	**28.63**	**27.62**	**27.60**
	SSIM↑	0.973	**0.977**	0.980	**0.987**	0.981	**0.981**	**0.975**	**0.981**	**0.979**	**0.911**	**0.897**	**0.883**	**0.897**
	LPIPS↓	**0.018**	**0.013**	**0.015**	0.022	0.021	**0.027**	**0.024**	**0.015**	**0.019**	0.141	**0.057**	**0.099**	**0.099**

Table 1: Quantitative evaluation of photometric fidelity on Replica and TUM RGB-D datasets.

5 Experiments

5.1 Experimental Setup

Implementation

GGF-SLAM is implemented using the LibTorch C++ framework with custom CUDA kernels for differentiable rasterization. All experiments are conducted on a desktop PC equipped with an Intel Core i9-13900K CPU and a single NVIDIA RTX 4090 GPU. For the GGF module, the probability threshold τ_{thresh} is set to 0.6 to distinguish between planar and volumetric primitives. The weights for the optimization objective in Eq. 6 are empirically set to $\lambda_{photo} = 1.0$, $\lambda_{depth} = 0.5$, $\lambda_{normal} = 0.05$, and $\lambda_{morph} = 0.1$. Furthermore, to ensure consistent appearance modeling across different viewpoints and high-quality reconstruction of high-frequency details, the AfME strategies proposed in [Wen *et al.*, 2025] are adopted.

Datasets & Baselines

We evaluate GGF-SLAM on three standard datasets: Replica [Straub *et al.*, 2019], TUM RGB-D [Sturm *et al.*, 2012], and ScanNet [Dai *et al.*, 2017]. For reconstruction quality (photometric and geometric), comparisons are conducted on Replica and TUM RGB-D against neural implicit methods [Zhu *et al.*, 2022; Wang *et al.*, 2023] as well as state-of-the-art 3DGS-based frameworks, including [Yan *et al.*, 2024; Keetha *et al.*, 2024; Huang *et al.*, 2024b; Matsuki *et al.*, 2024; Wen *et al.*, 2025; Ha *et al.*, 2024]. For camera tracking accuracy, we benchmark our method on TUM RGB-D and ScanNet against NICE-SLAM, Co-SLAM, and the aforementioned Gaussian approaches. In all subsequent quantitative tables, the best results are marked as best, second and third. '-' denotes that the system does not provide valid results.

5.2 Results and Analysis

Reconstruction Quality

In this section, both photometric fidelity and geometric accuracy are evaluated to validate the effectiveness of the proposed GGF module. For photometric fidelity, the standard rendering quality metrics, including PSNR, SSIM, and LPIPS [Zhang *et al.*, 2018] are reported. For geometric accuracy, Depth L1 is selected as the evaluation metric. As summarized in Table 1, GGF-SLAM outperforms SOTA approaches in terms of average results across all metrics on both synthetic and real-world datasets. Notably, on the Replica dataset, we achieve a mean PSNR of 40.51 dB, surpassing the strong baselines SEGS-SLAM (39.42 dB) and SplaTAM (33.85 dB). Crucially, this photometric gain is not achieved by overfitting texture, but by strictly adhering to the underlying spatial structure. This is evidenced by Table 2, where GGF-SLAM achieves the lowest average Depth L1 error (0.52 cm), demonstrating a 25% improvement over SEGS-SLAM (0.68 cm).

Method	R0	R1	R2	Of0	Of1	Of2	Of3	Of4	Avg.
NICE-SLAM[Zhu *et al.*, 2022]	1.79	1.33	2.20	1.43	1.58	2.70	2.10	2.06	1.90
Co-SLAM[Wang *et al.*, 2023]	1.05	0.85	2.37	1.24	1.48	1.86	1.66	1.54	1.51
GS-SLAM[Yan *et al.*, 2024]	1.31	0.82	1.26	0.81	0.96	1.41	1.53	1.08	1.15
SplaTAM[Keetha *et al.*, 2024]	**0.43**	**0.38**	0.54	**0.44**	0.66	1.05	1.60	**0.68**	0.72
SEGS-SLAM*[Wen *et al.*, 2025]	0.80	0.58	0.63	0.47	0.34	0.86	0.97	0.79	0.68
Ours	0.46	0.42	**0.50**	**0.44**	**0.30**	**0.65**	**0.64**	0.70	**0.51**

Table 2: Geometry reconstruction (Depth L1↓ [cm]) on Replica dataset. SEGS-SLAM* denotes the extension of the depth rendering pipeline.

The visual comparison in Figure 5 further proves the superiority of GGF-SLAM. Existing methods like MonoGS often suffer from ambiguity between geometry and appearance, causing Gaussians to overfit texture as unstructured blobs, which results in blurry structural edges and noisy depth maps (e.g., the table leg in Figure 5). Leveraging the adaptive regulation of GGF, GGF-SLAM seamlessly transitions between planar disks for surfaces and volumetric ellipsoids for unstructured regions, which eliminates floating artifacts, yielding sharp RGB edges and high-fidelity depth maps.

Camera Tracking Accuracy

In this section, quantitative comparisons of localization evaluation on the TUM RGB-D and ScanNet datasets are performed. Consistent with [Wen *et al.*, 2025], the root mean

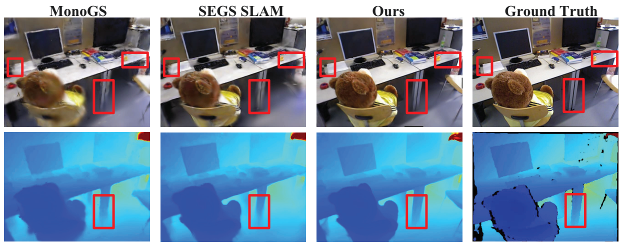


Figure 5: Qualitative comparison on TUM RGB-D. Red insets highlight fine structures.

square error (RMSE) of the absolute trajectory error (ATE) is reported. As shown in Table 3, GGF-SLAM achieves state-of-the-art performance with the lowest average RMSE of 0.94 cm on the TUM RGB-D dataset, significantly outperforming recent competitive methods such as SplaTAM (3.26 cm), MonoGS (1.50 cm), and the baseline SEGS-SLAM (1.53 cm). Crucially, our system demonstrates exceptional robustness in challenging texture-less scenarios, such as the fr3/o sequence. While pure photometric-based optimization methods often drift in such regions due to the lack of distinct visual features (evident in SplaTAM's error of 3.33 cm), our method effectively limits the tracking error to 0.96 cm. Furthermore, experimental results on the ScanNet dataset demonstrate the generalization capability of our method. We achieve the best average performance with an RMSE of 7.42 cm. Unlike frameworks relying solely on RGB residuals, our system exploits the high-fidelity depth and normal maps rendered by GGF to impose dense geometric constraints, anchoring the camera pose even when visual textures are sparse or ambiguous.

Method	TUM RGB-D				ScanNet						
	fr1/d	fr2/x	fr3/o	Avg.	0000	0059	0106	0169	0181	0207	Avg.
GS-SLAM[Yan *et al.*, 2024]	3.30	1.30	6.60	3.73	-	-	-	-	-	-	-
GS-ICP SLAM[Ha *et al.*, 2024]	3.54	2.25	2.97	2.92	-	-	-	-	-	-	-
SEGS-SLAM[Wen *et al.*, 2025]	3.19	0.37	1.03	1.53	-	-	-	-	-	-	-
MonoGS[Matsuki *et al.*, 2024]	1.53	1.44	1.54	1.50	15.94	**6.41**	19.44	10.44	12.23	10.46	12.49
Photo-SLAM[Huang *et al.*, 2024b]	3.56	**0.34**	1.70	1.87	7.62	7.94	9.36	10.01	22.97	7.23	10.86
RTG-SLAM [Peng *et al.*, 2024]	1.58	0.38	1.00	0.99	8.04	6.82	9.22	10.15	24.36	9.25	11.31
SplaTAM [Keetha *et al.*, 2024]	5.10	1.34	3.33	3.26	12.83	10.10	17.72	12.08	11.10	7.46	11.88
Ours	**1.52**	0.35	**0.96**	**0.94**	**6.21**	7.60	8.84	**6.93**	**8.34**	**6.58**	**7.42**

Table 3: Localization evaluation (RMSE↓ [cm]) on TUM RGB-D and ScanNet datasets.

System Efficiency and Robustness

As shown in Table 4, our method achieves real-time tracking (14.11 FPS) and rendering (425 FPS) on an RTX 4090, performing significantly faster than MonoGS and SplaTAM. With only a marginal memory overhead compared to SEGS-SLAM (4.73 GB vs. 4.10 GB peak), it demonstrates a superior efficiency-accuracy trade-off. Furthermore, the threshold τ_{thresh} maintains a stable performance plateau within the range of $[0.5, 0.7]$.

5.3 Ablation Studies

In this section, the effectiveness of the proposed core components is validated. As shown in Table 5, removing GGF leads to a significant increase in Depth L1 error, while disabling GCGR causes the RMSE of ATE to rise, indicating tracking

Metric	MonoGS	Photo-SLAM	RTG-SLAM	SplaTAM
Rendering ↑	706	1562	447	531
Tracking ↑	1.33	30.30	17.24	0.15
Metric	SGS-SLAM	GS-ICP SLAM	SEGS-SLAM	**Ours**
Rendering ↑	486	630	400	**425**
Tracking ↑	0.14	30.32	17.18	**14.11**

Table 4: Runtime comparison (FPS) of different SLAM systems.

instability. Visually, Figure 6 demonstrates severe artifacts on planar surfaces without GGF, confirming that GGF is essential for structural regularization and floater elimination. Our full method achieves the best trade-off between geometric fidelity and tracking accuracy.

Modules			R0 (Replica)				
Base	GGF	GCGR	PSNR↑	SSIM↑	LPIPS↓	Depth $L1$↓	RMSE↓
✓			37.07	0.968	0.023	0.80	0.296
✓	✓		37.73	0.968	0.20	0.52	0.301
✓	✓	✓	**38.17**	**0.973**	**0.018**	**0.46**	**0.275**
Modules			**fr1/d (TUM RGB-D)**				
Base	GGF	GCGR	PSNR↑	SSIM↑	LPIPS↓	Depth $L1$↓	RMSE↓
✓			25.29	0.839	0.136	5.12	3.187
✓	✓		26.23	0.840	0.136	4.86	3.055
✓	✓	✓	**26.56**	**0.911**	**0.141**	**4.34**	**1.524**

Table 5: Ablation study of different modules on two sequences.

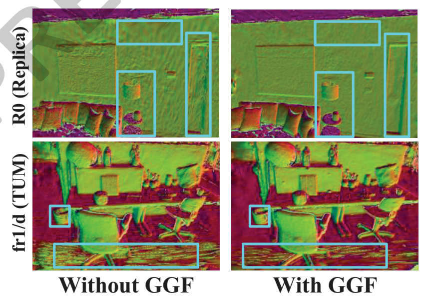


Figure 6: Reconstruction comparison of whether to use GGF, which can produce more complete and accurate spatial structures.

6 Conclusion

This paper presents GGF-SLAM, a novel tightly-coupled framework that effectively bridges the gap between Gaussian Splatting and SLAM. Through the proposed GGF, explicit structural priors are extracted from the SLAM map and fused with implicit neural residuals to adaptively regulate Gaussian morphologies, achieving a balance between geometric consistency and photorealistic rendering. Furthermore, GCGR exploits dense geometric constraints derived from Gaussian primitives to robustly rectify camera poses. Consequently, a bidirectional mutual enhancement and tight-coupling mechanism between tracking and mapping is established. Extensive evaluations on standard benchmarks demonstrate that GGF-SLAM significantly outperforms state-of-the-art methods in tracking accuracy, appearance fidelity, and geometric reconstruction quality, validating the efficacy of integrating structural constraints into Gaussian primitives. For future work, the utilization of Gaussian primitives to implement loop closure in SLAM will be investigated, thereby further enhancing localization and mapping performance.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grants 2024YFB4708900, in part by the Natural Science Foundation of China under Grant 62473208 and U21A20486, and in part by the Fundamental Research Funds for the Central Universities under Grant Number 63243158 and 63253238.

References

[Campos *et al.*, 2021] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

[Dai *et al.*, 2017] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[Davison *et al.*, 2007] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[Engel *et al.*, 2014] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[Engel *et al.*, 2017] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[Forster *et al.*, 2014] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[Ha *et al.*, 2024] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgbd gs-icp slam. In *European Conference on Computer Vision*, pages 180–197. Springer, 2024.

[Hsiao *et al.*, 2017] Ming Hsiao, Eric Westman, Guofeng Zhang, and Michael Kaess. Keyframe-based dense planar slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5110–5117. Ieee, 2017.

[Huang *et al.*, 2024a] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024.

[Huang *et al.*, 2024b] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. 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*, pages 21584–21593, 2024.

[Kaess, 2015] Michael Kaess. Simultaneous localization and mapping with infinite planes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4605–4611. IEEE, 2015.

[Keetha *et al.*, 2024] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024.

[Kerbl *et al.*, 2023] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[Klein and Murray, 2007] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[Li *et al.*, 2024] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *European Conference on Computer Vision*, pages 163–179. Springer, 2024.

[Liso *et al.*, 2024] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R Oswald. Loopy-slam: Dense neural slam with loop closures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20363–20373, 2024.

[Lu *et al.*, 2024] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024.

[Matsuki *et al.*, 2024] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024.

[Mur-Artal and Tardós, 2017] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[Mur-Artal *et al.*, 2015] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[Newcombe *et al.*, 2011] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011.

[Peng *et al.*, 2024] Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024.

[Pumarola *et al.*, 2017] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. Pl-slam: Real-time monocular visual slam with points and lines. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4503–4508. IEEE, 2017.

[Straub *et al.*, 2019] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[Sturm *et al.*, 2012] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[Sucar *et al.*, 2021] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6229–6238, 2021.

[Taguchi *et al.*, 2013] Yuichi Taguchi, Yong-Dian Jian, Srikumar Ramalingam, and Chen Feng. Point-plane slam for hand-held 3d sensors. In *2013 IEEE international conference on robotics and automation*, pages 5182–5189. IEEE, 2013.

[Wang *et al.*, 2023] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. 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*, pages 13293–13302, 2023.

[Wen *et al.*, 2025] Tianci Wen, Zhiang Liu, and Yongchun Fang. Segs-slam: Structure-enhanced 3d gaussian splatting slam with appearance embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28103–28113, 2025.

[Whelan *et al.*, 2015] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. In *Robotics: science and systems*, volume 11. Rome, 2015.

[Xin *et al.*, 2024] Zhe Xin, Yufeng Yue, Liangjun Zhang, and Chenming Wu. Hero-slam: Hybrid enhanced robust optimization of neural slam. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8610–8616. IEEE, 2024.

[Yan *et al.*, 2024] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024.

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[Zhang *et al.*, 2023] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023.

[Zhu *et al.*, 2022] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022.