

SEGS-SLAM: Structure-enhanced 3D Gaussian Splatting SLAM with Appearance Embedding

Tianci Wen^{1,2} Zhiang Liu^{1,2} Yongchun Fang^{1,2}*

¹ IRAIS, tjKLIR, College of Artificial Intelligence, Nankai University

² IITRS, Shenzhen Research Institute, Nankai University

<https://segs-slam.github.io/>

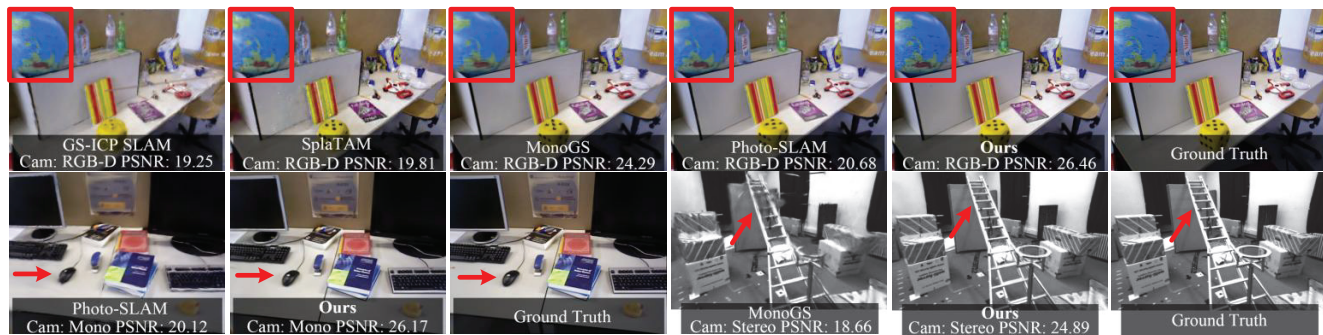


Figure 1. Our method SEGS-SLAM outperforms SOTA methods (GS-ICP SLAM [11], Photo-SLAM [14], SplaTAM [16], MonoGS [26]) in photorealistic mapping quality across monocular, stereo, and RGB-D cameras. The scenes are from TUM RGB-D dataset [34] (the top row and the left three images in the bottom row) and the EuRoC MAV dataset [2] (the right three images in the bottom row).

Abstract

3D Gaussian splatting (3D-GS) has recently revolutionized novel view synthesis in the simultaneous localization and mapping (SLAM) problem. However, most existing algorithms fail to fully capture the underlying structure, resulting in structural inconsistency. Additionally, they struggle with abrupt appearance variations, leading to inconsistent visual quality. To address these problems, we propose SEGS-SLAM, a structure-enhanced 3D Gaussian Splatting SLAM, which achieves high-quality photorealistic mapping. Our main contributions are two-fold. First, we propose a structure-enhanced photorealistic mapping (SEPM) framework that, for the first time, leverages highly structured point cloud to initialize structured 3D Gaussians, leading to significant improvements in rendering quality. Second, we propose Appearance-from-Motion embedding (AfME), enabling 3D Gaussians to better model image appearance variations across different camera poses. Extensive experiments on monocular, stereo, and RGB-D datasets demonstrate that SEGS-SLAM significantly outperforms state-of-the-art (SOTA) methods in photorealistic mapping quality,

e.g., an improvement of 19.86% in PSNR over MonoGS on the TUM RGB-D dataset for monocular cameras.

1. Introduction

Visual simultaneous localization and mapping (SLAM) is a fundamental problem in 3D computer vision, with wide applications in autonomous driving, robotics, virtual reality, and augmented reality. SLAM aims to construct dense or sparse maps to represent the scene. Recently, neural radiance fields (NeRF) [28] has been integrated into SLAM pipelines, significantly enhancing scene representation capabilities. The latest advancement in radiance field rendering is 3D Gaussian splatting (3D-GS) [17], an explicit scene representation that achieves revolutionary improvements in rendering and training speed. Recent SLAM works [11, 13, 14, 16, 19, 26, 29, 41, 42] incorporating 3D-GS have demonstrated that explicit representations provide more promising rendering performance when compared with implicit ones.

However, most SLAM algorithms based on 3D-GS have neglected the latent structure in the scene, which constrains their rendering quality. While some methods [11, 13, 16,

* Corresponding author.

[19, 26, 29, 41, 42, 45] have explored improvements in accuracy, efficiency, and semantics, insufficient structural exploitation remains a critical issue. For instance, as evidenced by the 2nd row in Fig. 1, MonoGS [26] produces a highly disorganized reconstruction of the ladder structure due to this limitation. In contrast, few methods, like Photo-SLAM [14], leverage scene structure. Photo-SLAM [14] initializes 3D Gaussians with point cloud obtained from indirect visual SLAM and incorporates a geometry-based densification module. In the original 3D-GS, 3D Gaussians are initialized from COLMAP [32] points. Since indirect visual SLAM and COLMAP [32] share similar pipeline structures, the generated point clouds exhibit similar intrinsic properties. Hence, the 3D Gaussians of Photo-SLAM [14] converge to a relatively optimal result with fewer iterations, yet it still underutilizes the underlying scene structure. The blurry reconstruction of the mouse edge by Photo-SLAM [14] is still apparent as shown in the 2nd row of Fig. 1.

Another partially unresolved challenge in these methods [11, 13, 14, 16, 19, 26, 29, 41, 42, 45] is the significant appearance variations within the scene (*e.g.*, exposure, lighting). To address this issue, NeRF-W [25] refines the appearance embeddings (AE) in NRW [27] and introduces them into NeRF. However, AE has a notable limitation: its training involves each ground-truth image from the test set. In novel view synthesis tasks, the test set contains 12.5% of all views, whereas in SLAM tasks, this increases to 80%, making it more challenging for AE to accurately predict appearance in novel views. Additionally, these approaches [11, 13, 14, 16, 19, 23, 26, 29, 41, 42, 45] fail to capture high-frequency details (*e.g.*, object edges, complex texture regions). FreGS [47] combines frequency regularization to model the local details, but its effectiveness is constrained by the use of a single-scale frequency spectrum.

To address the above limitations, this paper presents SEGS-SLAM, a novel 3D Gaussian Splatting SLAM system. First, we investigate the benefits of leveraging scene structure for improving rendering accuracy. While point cloud produced by ORB-SLAM3 [3] preserves strong latent structure, we observe that the anchor-based 3D Gaussians in [23] effectively leverage the underlying structure. Motivated by this, we propose a structure-enhanced photorealistic mapping (SEPM) framework, which initializes anchor points using ORB-SLAM3 [3] point cloud, significantly enhancing the utilization of scene structure. Experimental results validate the effectiveness of this simple yet powerful strategy, and we hope this insight will inspire further research in this direction. Second, we propose Appearance-from-Motion embedding (AfME), which takes poses as input and eliminates the need for training on the left half of each ground-truth image in the test set. We further introduce a frequency pyramid regularization (FPR) technique to better capture high-frequency details in the scene. The

main contributions of this work are as follows:

1. To our knowledge, structure-enhanced photorealistic mapping (SEPM) is the first SLAM framework that initializes anchor points with ORB-SLAM3 point cloud to strengthen the utilization of scene structure, leading to significant rendering improvements.
2. We propose Appearance-from-Motion embedding (AfME), which models per-image appearance variations into a latent space extracted from camera pose.
3. Extensive evaluations on various public datasets demonstrate that our method significantly surpasses state-of-the-art (SOTA) methods in photorealistic mapping quality across monocular, stereo, and RGB-D cameras, while maintaining competitive tracking accuracy.

2. Related Work

Visual SLAM. Traditional visual SLAM methods can be classified into two categories: indirect methods and direct methods. Indirect methods [3, 18] rely on extracting and tracking features between consecutive frames to estimate poses and build sparse maps by minimizing a reprojection error, including ORB-SLAM3 [3]. Direct methods [8, 9] estimate motion and structure by minimizing a photometric error, which can build sparse or semi-dense maps. Recently, some methods [1, 20, 35, 37, 38] have integrated deep learning into visual SLAM systems. Among them, the current SOTA method is Droid-SLAM [37]. More recently, Lipson *et al.* [20] combine optical flow prediction with a pose-solving layer to achieve camera tracking. Our approach favors traditional indirect visual SLAM.

Implicit Representation based SLAM. iMAP [36] pioneers the use of neural implicit representations to achieve tracking and mapping through reconstruction error. Subsequently, many works [4, 6, 7, 15, 24, 39, 43, 44, 51–53] have explored new representation forms, including voxel-based neural implicit surface representation [43], multi-scale tri-planes [15] and point-based neural implicit representation [30]. Recently, SNI-SLAM [51] and IBD-SLAM [44] introduce a hierarchical semantic representation and an xyz-map representation, respectively. Some works [5, 12, 21, 31, 46, 49, 50] address other challenges, including loop closure [21, 50]. However, most efforts focus on geometry reconstruction, with Point-SLAM [30] extending to novel view synthesis. Moreover, these NeRF-based methods do not account for appearance variations.

3D Gaussian Splatting based SLAM. Recently, an explicit representation, 3D-GS [17], is introduced into visual SLAM. Most methods enhance RGB-D SLAM in rendering quality [11, 16, 42, 45], efficiency [11, 29], robustness [13, 41], and semantics [19]. For example, GS-ICP SLAM [11] achieves photorealistic mapping by fusing 3D-GS with Generalized ICP. Few methods improve SLAM rendering accuracy and efficiency across monocular, stereo,

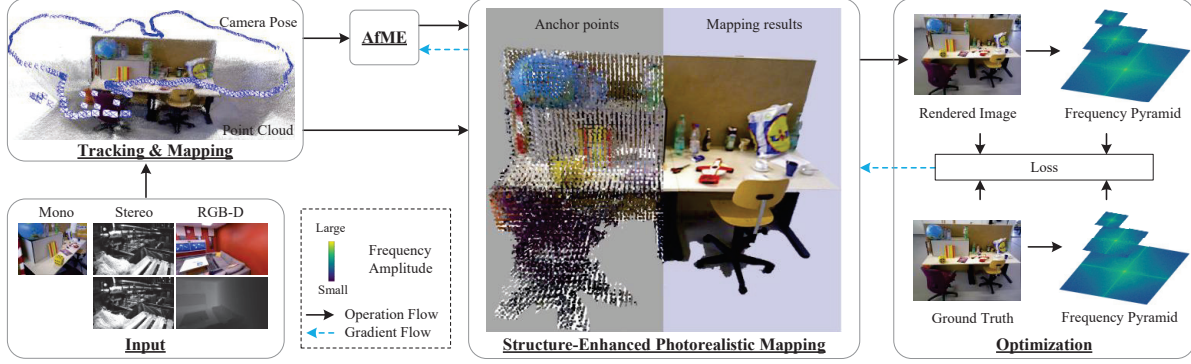


Figure 2. Overview of our method. Our method supports monocular, stereo, and RGB-D cameras. The input image stream is processed by the localization and geometric mapping modules, generating point cloud and accurate poses. SEPM incrementally initializes anchor points (middle) based on the point cloud (top left), which preserves the underlying structure. The poses are then fed into the AfME to model appearance variations in the scene. Additionally, we introduce FPR to improve the reconstruction of high-frequency details in the scene.

and RGB-D cameras, with MonoGS [26] and Photo-SLAM [14] as exceptions. However, neither of these methods effectively leverages latent scene structures or enhances the modeling of scene details, which limits their rendering quality. To address this, our proposed SEGS-SLAM further reinforces the utilization of global scene structure.

3. Preliminaries

In this section, we first introduce the structured 3D Gaussian splatting in [23]. Subsequently, we review the localization and mapping process of ORB-SLAM3 [3].

3.1. Structured 3D Gaussian Splatting

Structured 3D Gaussians is a hierarchical representation in [23]. They construct anchor points by voxelizing the point cloud obtained from COLMAP [32]. An anchor point is the center \mathbf{t}_v of a voxel, equipped with a context feature $\hat{f}_v \in \mathbb{R}^{32}$, a scale factor $l_v \in \mathbb{R}^3$, and a set of k learnable offsets $\mathbf{O}_v = \{\mathcal{O}_0, \dots, \mathcal{O}_{k-1}\} \in \mathbb{R}^{k \times 3}$. For each anchor point \mathbf{t}_v , they generate k 3D Gaussians, whose positions are calculated as:

$$\{\mu_0, \dots, \mu_{k-1}\} = \mathbf{t}_v + \{\mathcal{O}_0, \dots, \mathcal{O}_{k-1}\} \cdot l_v. \quad (1)$$

Other parameters of k 3D Gaussians are decoded using individual MLPs, denoted as M_α , M_c , M_q , and M_s , respectively. The colors of the Gaussians are obtained as follows:

$$\{c_0, \dots, c_{k-1}\} = M_C(\hat{f}_v, \delta_{vc}, \vec{\mathbf{d}}_{vc}), \quad (2)$$

where $\delta_{vc} = \|\mathbf{t}_v - \mathbf{t}_c\|_2$ is the relative distance between camera position \mathbf{t}_c and an anchor point and $\vec{\mathbf{d}}_{vc} = (\mathbf{t}_v - \mathbf{t}_c) / \delta_{vc}$ is their viewing direction. The opacity $\{\alpha_i\}$, quaternion $\{q_i\}$, and scale $\{s_i\}$ are similarly obtained. After obtaining the parameters of each 3D Gaussian $G(x)$ within the view frustum, it is projected onto the image plane to form

a 2D Gaussian $G'_i(x')$. Following 3D-GS [17], a tile-based rasterizer is used to sort the 2D Gaussians, and α -blending is employed to complete the rendering:

$$C(x') = \sum_{i \in N} c_i \delta_i \prod_{j=1}^{i-1} (1 - \delta_j), \quad \delta_i = \alpha_i G'_i(x'), \quad (3)$$

where x' is the pixel position and N is the number of corresponding 2D Gaussians for each pixel.

3.2. Localization and Geometry Mapping

ORB-SLAM3 [3] can track camera poses and generate point cloud accurately. The camera pose is represented as (\mathbf{R}, \mathbf{t}) , where $\mathbf{R} \in \text{SO}(3)$ denotes orientation and $\mathbf{t} \in \mathbb{R}^3$ represents position. The camera poses (\mathbf{R}, \mathbf{t}) and the point cloud $\{\mathbf{P}_0, \dots, \mathbf{P}_\eta\} \in \mathbb{R}^{\eta \times 3}$ of the scene can be solved through local or global bundle adjustment (BA):

$$\{\mathbf{P}_m, \mathbf{R}_l, \mathbf{t}_l\} = \underset{\mathbf{P}_m, \mathbf{R}_l, \mathbf{t}_l}{\operatorname{argmin}} \sum_{\kappa \in \mathcal{K}_L \cup \mathcal{K}_F} \sum_{j \in \mathcal{X}_k} \rho(E(\kappa, j)), \quad (4)$$

$$E(\kappa, j) = \|\mathbf{p}_j - \pi(\mathbf{R}_\kappa \mathbf{P}_j + \mathbf{t}_\kappa)\|_{\Sigma_g}^2, \quad (5)$$

where $m \in P_L$, $l \in \mathcal{K}_L$, \mathcal{K}_L is a set of covisible keyframes, P_L are the points seen in \mathcal{K}_L , \mathcal{K}_F are other keyframes, \mathcal{X}_k is the set of matched points between a keyframe κ and P_L , ρ is the robust Huber cost function, $E(\kappa, j)$ is the reprojection error between the matched 3D points \mathbf{P}_j and 2D feature points \mathbf{p}_j , π is the projection function, and Σ_g denotes the covariance matrix associated with the keypoint's scale. We provide more details in Sec. 11.2 of supplementary material.

4. SEGS-SLAM

In this section, we first give an overview of our system. We then present details of two key innovations: SEPM and AfME. Finally, we provide an introduction to our training

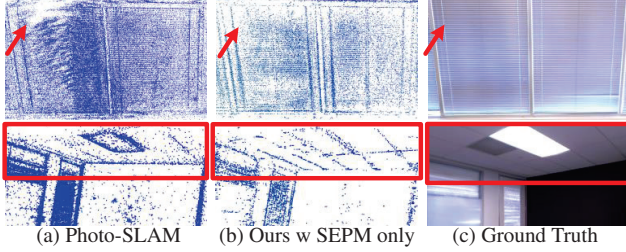


Figure 3. Visualization of the Photo-SLAM’s 3D Gaussians and of our method’s anchor points using only SEPM after 30k iterations. SEPM enhances the underlying structure of the 3D representation.

loss and FPR. The overview of our SEGS-SLAM is summarized in Fig. 2. First, the input image stream is processed through tracking and geometric mapping process in Sec. 3.2 to obtain camera poses and point cloud. On one hand, SEPM voxelizes the point cloud to initialize anchors, enhancing the exploitation of underlying scene structure. On the other hand, AfME encodes the camera poses to model appearance variations. Finally, the rendered images are supervised by ground-truth images, with the assistance of FPR, while jointly optimizing the parameters of the 3D Gaussians and the weights of AfME throughout training.

4.1. Structure-Enhanced Photorealistic Mapping

A common limitation of existing 3DGS-based SLAM systems is the gradual degradation of the underlying structure of Gaussians during optimization, limiting the quality of the rendered results. *Our key insight is that preserving strong scene structure throughout the optimization process is crucial for achieving high-fidelity rendering.* Inspired by this, we make several observations. First, ORB-SLAM3 [3] generates a point cloud that preserves the scene structure well. Photo-SLAM [14] initializes 3D Gaussians with the point cloud, but as shown in the Fig. 3 (a), relying solely on the latent structure of the point cloud is insufficient, as Gaussians still undergo structural degradation during optimization. We further observe that the hierarchical structure in [23] leverages scene structure, using anchor points to manage 3D Gaussians. These anchor points remain fixed during the optimization process, ensuring that 3D Gaussians retain the underlying structure of the scene.

Based on this observation, we propose incrementally voxelizing the point cloud P_k of each keyframe to construct anchor points, as follows:

$$V_k = \left\{ \left\lfloor \frac{P_k}{\epsilon} \right\rfloor \right\} \cdot \epsilon, \quad (6)$$

where $V_k \in R^{N \times 3}$ denotes voxel centers, ϵ is the voxel size, and $\{\cdot\}$ is the operation to remove redundant points.

Fig. 2 visualizes this process. The ORB-SLAM3 point cloud (top left) is voxelized into the anchor points (middle),

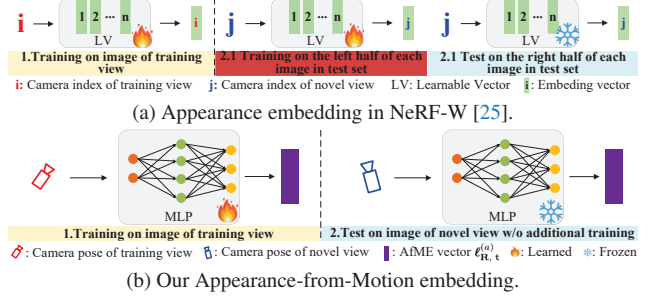


Figure 4. AE [25] and the proposed AfME. The differences between them are: (1) AE uses image indexes as input, whereas AfME leverages camera poses. (2) AE adopts learnable vectors (LV), while AfME utilizes a tiny MLP. (3) Most critically, our AfME requires no additional training on novel views.

which preserve the underlying structure effectively. In this way, the structural prior from ORB-SLAM3’s point cloud and the anchor-based organization are seamlessly fused, enhancing the exploitation of the underlying structure. Specifically, the anchor points inherit the strong structural properties of the point cloud and, due to their fixed positions, consistently preserve the latent scene structure throughout training. As shown in the Fig. 3 (b), this effectively prevents structural degradation of Gaussians over training, a key issue in prior methods. This strategy yields substantial improvements in rendering accuracy, as demonstrated in our ablation studies. Although conceptually simple, our approach proves highly effective. After anchor points construction, 3D Gaussians are then generated according to Eq. (1), Eq. (8), and rendered via Eq. (3).

4.2. Appearance-from-Motion Embedding

To further enhance the rendering quality of SEGS-SLAM, we observe a drop in the photorealistic mapping quality when suffering from appearance changes. AE is a proven solution in [23, 25] for handling such variations. However, the limitation of the AE in [25] lies in involving each ground-truth (left half of the image) in the test set for training as shown in the mid of Fig. 4 (a). To address this issue, we propose Appearance-from-Motion embedding (AfME), which employs a lightweight Multilayer Perceptron (MLP) M_{θ_a} to learn a shared appearance representation. As illustrated in Fig. 4 (b), the input of the encoder M_{θ_a} is the camera pose (\mathbf{R}, \mathbf{t}) . The MLP encodes the pose and outputs an embedding vector $\ell_{\mathbf{R}, \mathbf{t}}^{(a)}$, as shown below:

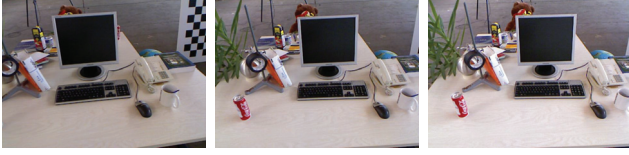
$$\ell_{\mathbf{R}, \mathbf{t}}^{(a)} = M_{\theta_a}(\mathbf{R}, \mathbf{t}). \quad (7)$$

Subsequently, the embedding vector $\ell_{\mathbf{R}, \mathbf{t}}^{(a)}$ is fed into the color decoder M_C . After introducing AfME, the color prediction of 3D Gaussians changes from Eq. (2) to:

$$\{c_0, \dots, c_{k-1}\} = M_C(\hat{f}_v, \delta_{vc}, \vec{\mathbf{d}}_{vc}, \ell_{\mathbf{R}, \mathbf{t}}^{(a)}). \quad (8)$$



(a) Same-view renderings with different appearance embedded by AfME.



(b) Renderings from different views correspond to different appearances.

Figure 5. The visualization of AfME controlling appearance. The rendering viewpoints in the top three images above are same, and only the input to AfME has been changed. The input poses of the AfME in the top-row images correspond to those in the bottom-row images. The results show that only the color and illumination have changed, while the geometry is fixed.

More details are in Sec. 11.4 of supplementary material. We choose camera poses as inputs for several reasons: 1) Similar to image indices, camera poses are unique for each view. 2) Camera poses naturally represent spatial information, enabling AfME to predict appearance from spatial context. 3) Camera poses are more continuous than image indices.

We adopt AfME to encode the scene appearance into the continuous pose space. Through training on the training set, AfME learns the mapping between appearance and camera pose, enabling it to predict the appearance for novel views. We conduct an experiment to demonstrate this. After training, we fix \hat{f}_v , δ_{vc} , \bar{d}_{vc} in the input of Eq. (8) and vary only the pose $\{\mathbf{R}, \mathbf{t}\}$ fed into AfME. As shown in Fig. 5, the lighting conditions under the same view change consistently with the pose $\{\mathbf{R}, \mathbf{t}\}$, matching the illumination of corresponding viewpoints. This demonstrates that the illumination conditions can be embedded by AfME effectively.

4.3. Frequency Pyramid Regularization

A minor improvement is the frequency pyramid regularization (FPR), which leverages multi-scale frequency representation to enhance the reconstruction of high-frequency details in the scene. To achieve this, we apply bilinear interpolation to downsample both the render images I_r and the ground truth images I_g . Let $s \in \mathcal{S} = \{s_0, s_1, \dots, s_n\}$ denote the scale of an image. We apply a 2D Fast Fourier Transform (FFT) to obtain the frequency spectra $\mathcal{F}(I_r^s)(u, v)$, $\mathcal{F}(I_g^s)(u, v)$. The loss \mathcal{L}_{hf} is computed as

$$\mathcal{L}_{hf} = \sum_{s \in \mathcal{S}} \frac{1}{N} \lambda_s \sum_{u, v} |F_{hf,r}^s(u, v) - F_{hf,g}^s(u, v)|, \quad (9)$$

$$F_{hf,i}^s(u, v) = H_{hf}(u, v) \cdot \mathcal{F}(I_i^s)(u, v), \quad i \in \{r, g\}, \quad (10)$$

where $F_{hf,r}^s(u, v)$, $F_{hf,g}^s(u, v)$ is the high-frequency extracted by a high-pass filter $H_{hf}(u, v)$, $N = HW$ denotes the image size, and λ_s represents the weight of each scale. More details are in Sec. 8 of supplementary material.

4.4. Losses Design

The optimization of the learnable parameters, the MLP M_α , M_c , M_q , M_s , and M_{θ_α} , are achieved by minimizing the L1 loss \mathcal{L}_1 , SSIM term [40] \mathcal{L}_{SSIM} , frequency regularization \mathcal{L}_{hf} , and volume regularization [22] \mathcal{L}_{vol} between the rendered images and the ground truth images, denoted as

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda(1 - \mathcal{L}_{SSIM}) + \lambda_{vol}\mathcal{L}_{vol} + \lambda_{hf}\mathcal{L}_{hf}. \quad (11)$$

Following [23], we also incorporate \mathcal{L}_{vol} .

5. Experiment

5.1. Experiment Setup

Implementation. Our SEGS-SLAM is fully implemented using the LibTorch framework with C++ and CUDA. The training and rendering of 3D Gaussians involves three key modules: SEPM, AfME, and FPR, operating as a parallel thread alongside the localization and geometric mapping process. SEGS-SLAM trains the 3D Gaussians using only keyframe images, point clouds, and poses, where keyframes are selected based on co-visibility. In each iteration, SEGS-SLAM randomly samples a viewpoint from the current set of keyframes. We use the images and poses of keyframes as the training set, while the remaining images and poses serve as the test set. Moreover, following FreGS [47], we activate FPR once the structure of anchor points stabilizes and terminate it based on the completion of anchor point densification. The scale level of FPR is set to 3. Except for the non-open-source GS-SLAM [42], all methods compared in this paper are run on the same machine using their official code. The machine is equipped with an NVIDIA RTX 4090 GPU and a Ryzen 5995WX CPU. By default, our method runs for 30K iterations. The voxel size ϵ is 0.001 m. For Eq. (11), we set $\lambda = 0.2$, $\lambda_{vol} = 0.01$, $\lambda_{hf} = 0.01$.

Baselines. We first list the baseline methods used to evaluate photorealistic mapping. For monocular and stereo cameras, we compare our method with Photo-SLAM [14], MonoGS [26], and Photo-SLAM-30K. For RGB-D cameras, we additionally include comparisons with RTG-SLAM [29], GS-SLAM [42], SplaTAM [16], SGS-SLAM [19], and GS-ICP SLAM [11], all of which represent SOTA SLAM methods based on 3D-GS. To ensure fairness, we set the maximum iteration limit to 30K for all methods, following the original 3D-GS [17]. Photo-SLAM-30K refers to Photo-SLAM [14] with a fixed iteration count of 30K. For camera pose estimation, we also compare ours with ORB-SLAM3 [3] and DROID-SLAM [37].

Datasets (Camera) Method	Replica (RGB-D)			TUM RGB-D (RGB-D)		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
MonoGS [26]	36.81	0.964	0.069	24.11	0.800	0.231
Photo-SLAM [14]	35.50	0.949	0.056	20.99	0.736	0.213
Photo-SLAM-30K	36.94	0.952	0.040	21.73	0.757	0.186
RTG-SLAM [29]	32.79	0.918	0.124	16.47	0.574	0.461
GS-SLAM* [42]	34.27	0.975	0.082	-	-	-
SplaTAM [16]	33.85	0.936	0.099	21.41	0.764	0.265
SGS-SLAM [19]	33.96	0.969	0.099	-	-	-
GS-ICP SLAM [11]	37.14	0.968	0.045	17.81	0.642	0.361
Ours	39.42	0.975	0.021	26.03	0.843	0.107

Table 1. Quantitative evaluation of our method compared to SOTA methods for **RGB-D** camera on Replica and TUM RGB-D datasets. Best results are marked as **best score**, **second best score** and **third best score**. GS-SLAM* denotes the result of GS-SLAM is taken from [42], all others are obtained in our experiments. '-' denotes the system does not provide valid results.

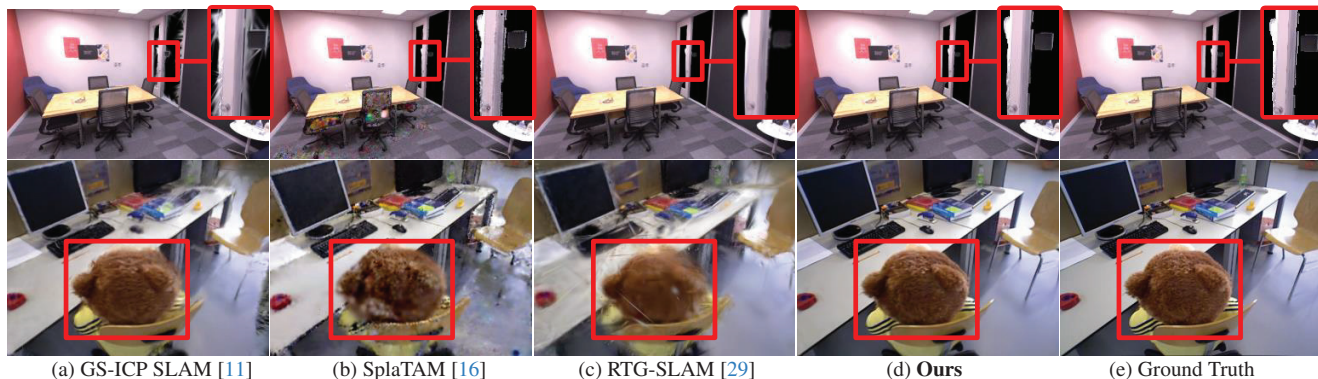


Figure 6. We show comparisons of ours to SOTA methods for **RGB-D** camera. The top scene is *office2* from the Replica datasets, and the bottom is *fr3/office* from TUM RGB-D datasets. Non-obvious differences in quality are highlighted by insets.

Metrics. We follow the evaluation protocol of MonoGS [26] to assess both camera pose estimation and novel view synthesis. For camera pose estimation, we report the root mean square error (RMSE) of the absolute trajectory error (ATE) [10] for all frames. For photorealistic mapping, we report standard rendering quality metrics, including PSNR, SSIM, and LPIPS [48]. To evaluate the photorealistic mapping quality, we only calculate the average metrics over novel views for all methods. We report the average across five runs for all methods. *To ensure fairness, no training views are included in the evaluation, and for all RGB-D SLAM methods, no masks are applied to either the rendered or ground truth images during metric calculation.* As a result, the reported metrics for Photo-SLAM [14] are slightly lower than those in the original paper, as they averages both novel and training views. Similarly, the metrics of SplaTAM [16], SGS-SLAM [19], and GS-ICP SLAM [11] are slightly lower than reported, as the original methods use a mask to exclude outliers for both the rendered and ground truth images based on anomalies in the depth image.

Datasets. Following [11, 13, 14, 16, 19, 26, 29, 30, 36, 42], we evaluate all methods on all sequences of the Replica dataset [33] for monocular and RGB-D cameras. Following

[11, 14, 26, 29, 36, 42, 52], we use the *fr1/desk*, *fr2/xyz*, and *fr3/office* sequences of the TUM RGB-D dataset [34] for monocular and RGB-D cameras. Following [14], we use the *MH01*, *MH02*, *V101*, and *V201* sequences of the EuRoC MAV dataset [2] for stereo cameras.

5.2. Results Analysis

Camera Tracking Accuracy. As shown in Tab. 3, our method demonstrates competitive accuracy in tracking for monocular, stereo, and RGB-D cameras when compared with SOTA methods. This highlights the advantage of indirect visual SLAM in terms of localization accuracy.

Novel View synthesis. The quantitative rendering results for novel views in *RGB-D* scenarios are shown in Tab. 1, where SEGS-SLAM significantly outperforms comparison methods, achieving the highest average rendering quality on both TUM RGB-D and Replica datasets. In the top of Fig. 6, it is evident that for the Replica dataset, only our method can accurately recover the contours of edge regions. The TUM RGB-D dataset presents a greater challenge compared with the Replica dataset, with highly cluttered scene structures and substantial lighting variations. GS-ICP SLAM [11], a leading RGB-D SLAM method

Datasets (Camera) method	Replica (Mono)			TUM RGB-D (Mono)			EuRoC (Stereo)		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
MonoGS [26]	28.34	0.878	0.256	21.00	0.705	0.393	22.60	0.789	0.274
Photo-SLAM [14]	33.60	0.934	0.077	20.17	0.708	0.224	11.90	0.409	0.439
Photo-SLAM-30K	36.08	0.947	0.054	21.06	0.733	0.186	11.77	0.405	0.430
Ours	37.96	0.964	0.037	25.17	0.825	0.122	23.64	0.791	0.182

Table 2. Quantitative evaluation of our method compared to SOTA methods for **Monocular (Mono)** and **Stereo** cameras on Replica, TUM RGB-D, and EuRoC MAV datasets. Best results are marked as **best score** and **second best score**.

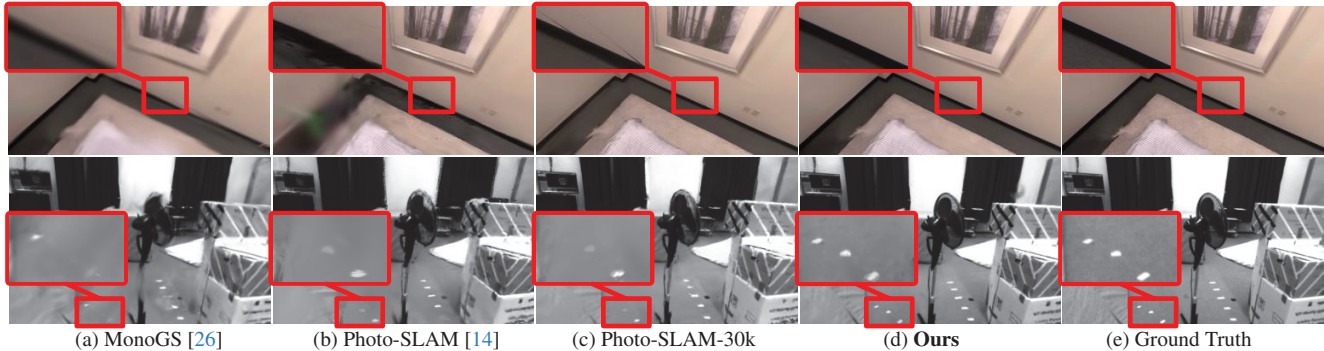


Figure 7. We show comparisons of ours to SOTA methods for **Monocular** and **Stereo** cameras. The top scene is *room1* from the Replica dataset, and the bottom is *V201* from the EuRoC MAV dataset. Non-obvious differences in quality are highlighted by insets.

Camera Type Datasets Method	RGB-D			Monocular			Stereo
	Replica RMSE \downarrow	TUM R RMSE \downarrow	Avg. RMSE \downarrow	Replica RMSE \downarrow	TUM R RMSE \downarrow	Avg. RMSE \downarrow	EuRoC RMSE \downarrow
ORB-SLAM3 [3]	1.780	2.196	1.988	51.744	46.004	48.874	10.907
DRIO-SLAM [37]	74.264	74.216	74.24	76.600	1.689	39.145	1.926
MonoGS [26]	0.565	1.502	1.033	37.054	4.009	63.437	49.241
Photo-SLAM [14]	0.582	1.870	1.226	0.930	1.539	1.235	11.023
RTG-SLAM [29]	0.191	0.985	0.581	-	-	-	-
GS-SLAM* [42]	0.500	3.700	2.100	-	-	-	-
SplaTAM [16]	0.343	4.215	2.279	-	-	-	-
SGS-SLAM [19]	0.365	-	-	-	-	-	-
GS-ICP SLAM [11]	0.177	2.921	1.549	-	-	-	-
Ours	0.430	1.528	0.979	0.833	1.505	1.169	7.462

Table 3. Camera tracking result on Replica, TUM RGB-D (TUM R), and EuRoC MAV datasets for Monocular, stereo, and RGB-D cameras. **RMSE of ATE** (cm) is reported. The best results are marked as **best score**, **second best score** and **third best score**. '-' denotes that the system does not provide valid results.

based on 3D-GS, achieves the second-highest rendering accuracy on the Replica dataset. However, it relies heavily on depth images. As shown in the bottom of Fig. 6, GS-ICP SLAM [11] and other methods perform poorly on the TUM RGB-D dataset. Our SEGS-SLAM better reconstructs scene structure and lighting variations, benefiting from our SEPM and the AfME.

Tab. 2 presents quantitative rendering results for *monocular* scenarios, where SEGS-SLAM surpasses other methods. Notably, SEGS-SLAM continues to significantly outperform comparison methods on the TUM RGB-D dataset.

Importantly, compared with RGB-D scenarios, MonoGS [26] experiences a sharp decline. The top of Fig. 7 further demonstrates that on the Replica dataset, our method effectively models high-frequency details more realistically in regions such as the edge of the wall.

Moreover, our method remains effective in *stereo* scenarios. The corresponding quantitative results for realistic mapping are recorded in Tab. 2, where our method achieves the highest rendering quality, surpassing the current SOTA method, MonoGS [26]. As shown in the bottom of Fig. 7, our approach better reconstructs the global structure and local details of the scene. We highlight that a key factor enabling our method to achieve superior performance across different camera types and datasets is the proposed SEPM. Its enhancement to rendering quality is highly generalizable, as further demonstrated in ablation studies.

5.3. Ablation Studies

Structure-Enhanced photorealistic mapping. To evaluate the impact of SEPM on photorealistic mapping metrics, we additionally train two variants of our method: one without SEPM, AfME, and FPR, and another without AfME and FPR. The variant without SEPM, AfME, and FPR directly uses the original 3D-GS [17]. As shown in Tab. 4, rows (1) and (2), introducing SEPM consistently yields significant improvements in rendering quality across all scenes. This validates the effectiveness of enhancing the exploitation of latent structure. Moreover, it demonstrates that SEPM has strong generalization across diverse camera in-

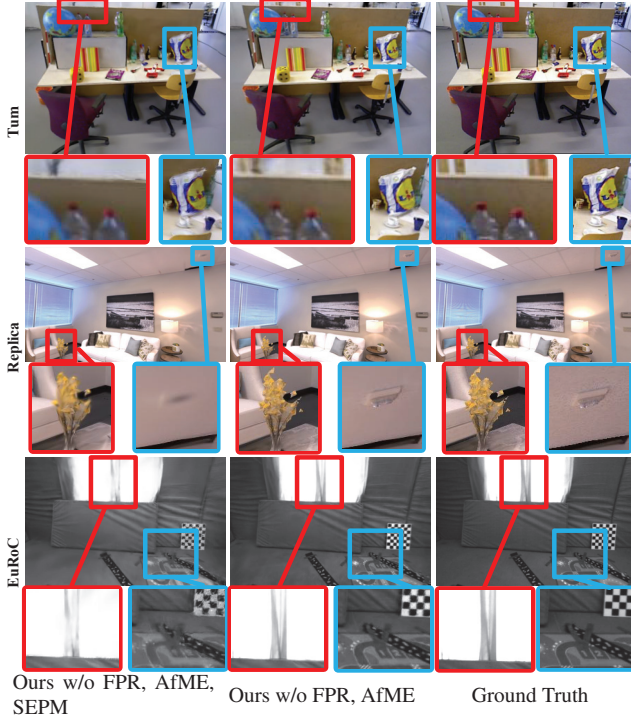


Figure 8. Ablation of SEPM. The figure presents qualitative results on the TUM, Replica, and EuRoC datasets. It is evident that, across all three datasets, incorporating SEPM enables the reconstruction of more complete and accurate structure.

puts and datasets, inspiring future research. Notably, with only SEPM, our rendering metrics surpass existing SOTA methods. To further illustrate the benefits of SEPM, Fig. 8 presents visual comparisons, where a more complete and accurate scene structure leads to superior rendering quality.

Appearance-from-Motion embedding. To evaluate the impact of the proposed AfME, we train an additional model for our method without AfME. In the rows (4) and (5) of Tab. 4, our full method (5) outperforms the model without AfME (4) in terms of PSNR scores. As shown in the top of Fig. 9, the AfME effectively predicts the lighting conditions of novel views. On the Replica dataset, the improvements from AfME are relatively modest. Replica is an easier dataset, in which PSNR already exceeds 37 without AfME, indicating that scene is well-reconstructed. Although incorporating AfME still yields a PSNR gain, the benefits are much more pronounced on the challenging TUM dataset, as clearly observed in both Fig. 9 and Tab. 4.

Frequency pyramid regularization. To evaluate the effect of the proposed FPR on photorealistic mapping metrics, we train an additional model for our method without FPR. As shown in Tab. 4, rows (3) and (5), our full method (5) surpasses the model without FPR (3) in terms of PSNR scores. Additionally, after applying the FPR, the model ren-

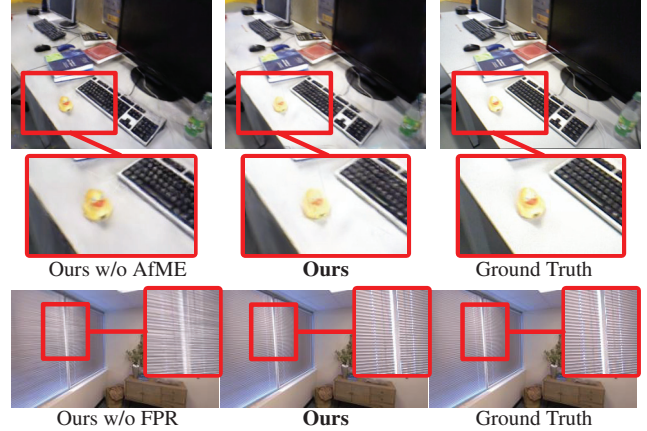


Figure 9. Ablation of AfME (*Top*) and FPR (*Bottom*). It is evident that with the introduction of AfME, the lighting conditions at novel views are accurately predicted, indicating the effectiveness of our AfME. The result without AfME is darker than the ground truth.

Camera type Datasets	RGB-D		Mono		Stereo
	Replica PSNR \uparrow	TUM R PSNR \uparrow	Replica PSNR \uparrow	TUM R PSNR \uparrow	EuRoC PSNR \uparrow
(1) w/o FPR, AfME, SEPM	36.07	21.73	35.46	21.10	11.76
(2) w/o FPR, AfME	38.98	24.20	36.31	23.54	22.91
(3) w/o FPR	39.18	25.04	36.44	24.91	23.52
(4) w/o AfME	39.12	24.66	37.48	23.69	22.99
(5) Ours	39.42	26.03	37.96	25.17	23.64

Table 4. Ablation Study on the **key components** (1) - (5). The best results are marked as **best score**.

ders finer details in highly textured regions, as demonstrated by the curtain at the bottom of Fig. 9.

5.4. Limitations

One limitation of our method is that a poorly structured point cloud leads to a decline in photorealistic mapping quality. Additionally, while our method achieves real-time tracking and rendering at 17 and 400 FPS, respectively, it exhibits reduced rendering speed due to the increased number of 3D Gaussians used to model high-frequency details. Currently, AFME is only capable of handling static scenes. By encoding more complex inputs, it can tackle more sophisticated dynamic scenarios.

6. Conclusion

We propose a novel SLAM system with progressively refined 3D-GS, termed SEGS-SLAM. Experimental results show that our method surpasses SOTA methods in rendering quality across monocular, stereo, and RGB-D datasets. We demonstrate that by enhancing the utilization of the underlying scene structure, SEPM improves the visual quality of rendering. Furthermore, our proposed AfME and FPR effectively predict the appearance of novel views and refine the scene details, respectively.

Acknowledgements: This work is supported by the National Natural Science Foundation of China under Grant 62233011.

References

- [1] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam — learning a compact, optimisable representation for dense visual slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 2
- [2] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016. 1, 6
- [3] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. 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. 2, 3, 4, 5, 7
- [4] Chi-Ming Chung, Yang-Che Tseng, Ya-Ching Hsu, Xiang-Qian Shi, et al. Orbeez-slam: A real-time monocular visual slam with orb features and nerf-realized mapping. In *2023 IEEE International Conference on Robotics and Automation*, pages 9400–9406, 2023. 2
- [5] Tianchen Deng, Guole Shen, Tong Qin, Jianyu Wang, Wentao Zhao, Jingchuan Wang, Danwei Wang, and Weidong Chen. Plgslam: Progressive neural scene representation with local to global bundle adjustment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19657–19666, 2024. 2
- [6] Tianchen Deng, Guole Shen, Xun Chen, Shenghai Yuan, Hongming Shen, Guohao Peng, Zhenyu Wu, Jingchuan Wang, Lihua Xie, Danwei Wang, Hesheng Wang, and Weidong Chen. Mcn-slam: Multi-agent collaborative neural slam with hybrid implicit neural scene representation. *arXiv preprint arXiv:2506.18678*, 2025. 2
- [7] Tianchen Deng, Guole Shen, Chen Xun, Shenghai Yuan, Tongxin Jin, Hongming Shen, Yanbo Wang, Jingchuan Wang, Hesheng Wang, Danwei Wang, et al. Mne-slam: Multi-agent neural slam for mobile robots. In *Conference on Computer Vision and Pattern Recognition*, pages 1485–1494, 2025. 2
- [8] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849, 2014. 2
- [9] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018. 2
- [10] Michael Grupp. evo: Python package for the evaluation of odometry and slam. <https://github.com/MichaelGrupp/evo>, 2017. 6
- [11] Seongbo Ha, Jiung Yeon, and Hyeonwoo Yu. Rgb-d gs-icp slam. In *European Conference on Computer Vision*, pages 180–197, 2024. 1, 2, 5, 6, 7, 4
- [12] Jiarui Hu, Mao Mao, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Cp-slam: Collaborative neural point-based slam system. In *Advances in Neural Information Processing Systems*, pages 39429–39442, 2023. 2
- [13] Jiarui Hu, Xianhao Chen, Boyin Feng, Guanglin Li, Liangjing Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field. In *European Conference on Computer Vision*, pages 93–112, 2024. 1, 2, 6
- [14] Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photo-realistic mapping for monocular stereo and rgb-d cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21584–21593, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- [15] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17408–17419, 2023. 2
- [16] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 1, 2, 5, 6, 7, 4
- [17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 1, 2, 3, 5, 7
- [18] 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, 2007. 2
- [19] 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, 2025. 1, 2, 5, 6, 7, 4
- [20] Lahav Lipson and Jia Deng. Multi-session slam with differentiable wide-baseline pose optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19626–19635, 2024. 2
- [21] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R. Oswald. Loopy-slam: Dense neural slam with loop closures. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20363–20373, 2024. 2
- [22] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics*, 40(4), 2021. 5
- [23] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2, 3, 4, 5, 1
- [24] Yunxuan Mao, Xuan Yu, Zhuqing Zhang, Kai Wang, Yue Wang, Rong Xiong, and Yiyi Liao. Ngel-slam: Neural

- implicit representation-based global consistent low-latency slam system. In *2024 IEEE International Conference on Robotics and Automation*, pages 6952–6958, 2024. 2
- [25] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2, 4
- [26] Hidenobu Matsuki, Riku Murai, Paul H.J. Kelly, and Andrew J. Davison. Gaussian splatting slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18039–18048, 2024. 1, 2, 3, 5, 6, 7, 4, 8, 9, 10
- [27] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rendering in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2
- [28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421, 2020. 1
- [29] 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. 1, 2, 5, 6, 7, 4
- [30] Erik Sandström, Yue Li, Luc Van Gool, and Martin R. Oswald. Point-slam: Dense neural point cloud-based slam. In *IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 2, 6
- [31] Erik Sandström, Kevin Ta, Luc Van Gool, and Martin R. Oswald. Uncle-slam: Uncertainty learning for dense neural slam. In *IEEE/CVF International Conference on Computer Vision Workshops*, pages 4537–4548, 2023. 2
- [32] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3
- [33] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6
- [34] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012. 1, 6
- [35] Edgar Sucar, Kentaro Wada, and Andrew Davison. Nodslam: Neural object descriptors for multi-view shape reconstruction. In *2020 International Conference on 3D Vision*, pages 949–958, 2020. 2
- [36] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. imap: Implicit mapping and positioning in real-time. In *IEEE/CVF International Conference on Computer Vision*, pages 6229–6238, 2021. 2, 6
- [37] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In *Advances in Neural Information Processing Systems*, pages 16558–16569, 2021. 2, 5, 7
- [38] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. In *Advances in Neural Information Processing Systems*, pages 39033–39051, 2023. 2
- [39] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Col-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023. 2
- [40] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 5
- [41] Yueming Xu, Haochen Jiang, Zhongyang Xiao, Jianfeng Feng, and Li Zhang. Dg-slam: Robust dynamic gaussian splatting slam with hybrid pose optimization. In *Advances in Neural Information Processing Systems*, 2024. 1, 2
- [42] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024. 1, 2, 5, 6, 7, 4
- [43] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality*, pages 499–507, 2022. 2
- [44] Minghao Yin, Shangzhe Wu, and Kai Han. Ibd-slam: Learning image-based depth fusion for generalizable slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10563–10573, 2024. 2
- [45] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 2
- [46] Hongjia Zhai, Gan Huang, Qirui Hu, Guanglin Li, Hujun Bao, and Guofeng Zhang. Nis-slam: Neural implicit semantic rgb-d slam for 3d consistent scene understanding. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–11, 2024. 2
- [47] Jiahui Zhang, Fangneng Zhan, Muyu Xu, Shijian Lu, and Eric Xing. Fregs: 3d gaussian splatting with progressive frequency regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21424–21433, 2024. 2, 5, 1, 3
- [48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 6
- [49] Wei Zhang, Tiecheng Sun, Sen Wang, Qing Cheng, and Norbert Haala. Hi-slam: Monocular real-time dense mapping with hybrid implicit fields. *IEEE Robotics and Automation Letters*, 9(2):1548–1555, 2024. 2
- [50] Youmin Zhang, Fabio Tosi, Stefano Mattoccia, and Matteo Poggi. Go-slam: Global optimization for consistent 3d instant reconstruction. In *IEEE/CVF International Conference on Computer Vision*, pages 3727–3737, 2023. 2

- [51] Siting Zhu, Guangming Wang, Hermann Blum, Jiuming Liu, Liang Song, Marc Pollefeys, and Hesheng Wang. Sni-slam: Semantic neural implicit slam. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21167–21177, 2024. [2](#)
- [52] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. [6](#)
- [53] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R. Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. In *International Conference on 3D Vision*, pages 42–52, 2024. [2](#)