2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

S^{3}LAM: Surfel Splatting SLAM for Geometrically Accurate Tracking and Mapping

Ruoyu Fan[1], Yu-Hui Wen[2], Tao Zhang[1], Long Zeng[3] and Yong-Jin Liu[1,*], *Senior Member, IEEE*

***Abstract*— We propose S^{3}LAM, a novel RGB-D SLAM system that leverages 2D surfel splatting to achieve geometrically accurate scene representations for simultaneous tracking and mapping. Unlike existing 3DGS-based SLAM approaches that rely on 3D Gaussian ellipsoids, we utilize 2D Gaussian surfels as primitives for more efficient scene representation. By focusing on the surfaces of objects in the scene, this design enables S^{3}LAM to reconstruct high-quality geometry, benefiting both mapping and tracking. To address inherent SLAM challenges including real-time optimization under limited viewpoints, we introduce a novel adaptive surface rendering strategy that improves mapping accuracy while maintaining computational efficiency. We further derive camera pose Jacobians directly from 2D surfel splatting formulation, highlighting the importance of our geometrically accurate representation that improves tracking convergence. Extensive experiments on both synthetic and real-world datasets demonstrate that S^{3}LAM achieves state-of-the-art performance. Our code is available at https://github.com/FanryZ/S3LAM.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a crucial component in various applications, such as autonomous driving, mobile robotics [1], and augmented reality. Over the past few decades, SLAM research has explored various representations for scene reconstruction, including point clouds [2], meshes [3], and voxel grids [4], [5], which have achieved good tracking and mapping performance.

Recently, representing scenes with neural radiance fields (NeRF) [6] has gained significant attention in SLAM research [7], [8]. These approaches combine view rendering and surface reconstruction with dense visual SLAM, expanding the potential applications of SLAM systems. Along this direction, a growing trend in visual SLAM is the use of 3D Gaussian Splatting (3DGS) [9] for scene mapping. Compared to radiance fields, 3DGS offers superior rendering speed, enhanced interpretability and flexible extensibility. 3DGS-based SLAM systems effectively integrate 3D Gaussian splatting techniques, incorporating innovations in Gaussian management [10], pose optimization [11], and uncertainty estimation [12]. These advancements demonstrate the significant potential of 3DGS for SLAM applications.

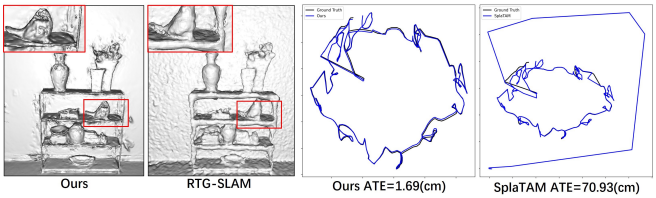


Fig. 1: Compared to state-of-the-art 3DGS-based SLAM systems [13], [14], our method achieves geometrically accurate scene reconstruction with fine details (Left vs. RTG-SLAM on *Room2*, Replica) and improved convergence in camera pose tracking (Right vs. SplaTAM on *S1*, ScanNet++).

Our work is motivated by the following observation: 3DGS-based SLAM methods reconstruct scenes using inverse rendering that prioritizes color consistency, which often compromises the reconstruction of fine geometric details such as surfaces. Since tracking and mapping are tightly coupled in SLAM, degraded mapping representation can in turn deteriorate tracking performance. The inherent characteristics of 3DGS constrain the SLAM performance in both mapping and tracking, as summarized below:

- **Mapping**: 3DGS reconstructs scene geometry by fitting the average depth of projected ellipsoids. Due to the lack of explicit surface normals, it struggles to accurately represent object surfaces, leading to visual artifacts and geometric inconsistencies across views.
- **Tracking**: Similarly, the absence of explicit surface normals in 3DGS restricts camera tracking to only optimize the value of Gaussian distribution functions, making it difficult for the estimated camera rotations to align with the true surface orientations in the scene.

To overcome the above limitations, in this paper, we adopt 2D Gaussian surfel splatting as scene representation primitives, inspired by the recent success of offline 2D Gaussian splatting approaches [15], [16]. For SLAM applications with limited Gaussian primitives, we introduce an adaptive surface reconstruction strategy that dynamically refines geometry. Furthermore, we present a novel pose estimation algorithm that explicitly leverages 2D surfels to robustly handle extreme viewpoint changes.

To sum up, we propose *Surfel Splatting SLAM* (S^{3}LAM), a novel system designed to achieve geometrically accurate mapping and tracking through 2D Gaussian surfel representations. S^{3}LAM integrates forward rendering for scene

This work was supported by the Natural Science Foundation of China (Project Number 62461160309).

[1]R. Fan, T. Zhang and Y.-J. Liu are with the Department of Computer Science, Tsinghua University, Beijing, China {fry21, zhang-t24}@mails.tsinghua.edu.cn, liuyongjin@tsinghua.edu.cn

[2]Y.-H. Wen is with the School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China yhwen1@bjtu.edu.cn

[3]Z. Long is with he Department of Advanced Manufacturing, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China zengl@sz.tsinghua.edu.cn

*Corresponding Author

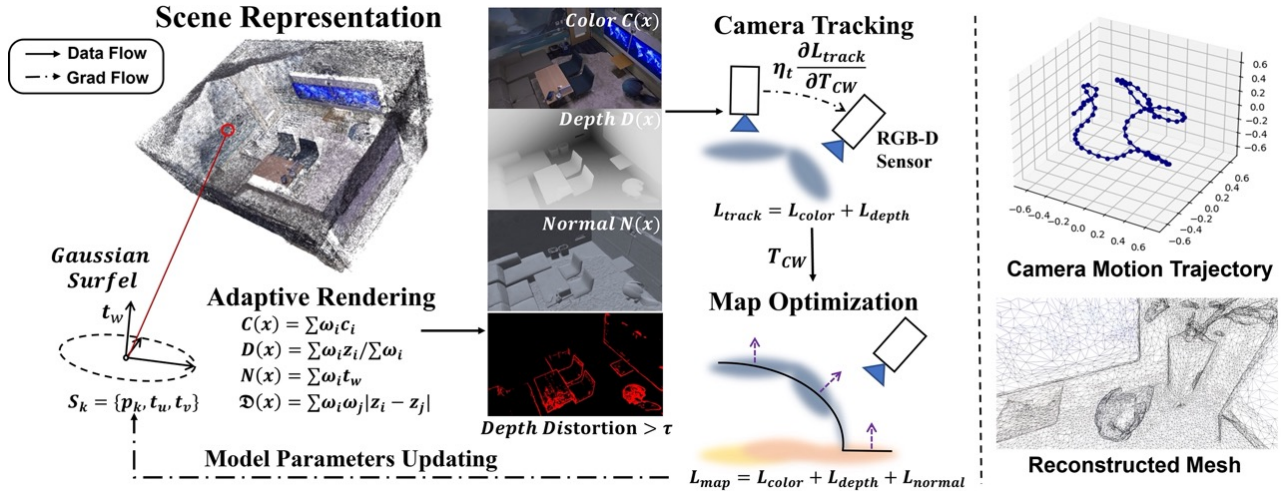


Fig. 2: **Overview of the proposed S^{3}LAM System.** The scene is represented by 2D Gaussian surfels to achieve geometry-aligned motion tracking and scene mapping. Both the tracking and mapping phases benefit from our oriented surfel primitives, leveraging the proposed adaptive rendering and surfel-based pose estimation. Our system outputs the motion trajectory and a reconstructed mesh model of the scene after post-processing.

reconstruction with backward gradient propagation to jointly optimize color, depth, and normal estimation. To enable online adaptive mapping, we first quantify the uncertainty in different regions by introducing a depth distortion term during optimization. This distortion term identifies areas under construction in mapping, triggering our adaptive reconstruction strategy that replaces uncertain geometry with dominant surfels. For camera tracking, we derive an analytic Jacobian on Lie algebra for the surfel splatting map. Our approach incorporates a radial gradient in the surfel-based Jacobian, enabling explicit alignment optimization between the camera orientation and reconstructed surfaces. These designs enable our method to achieve accurate geometric reconstruction and high-convergence pose tracking (Figure 1).

Our contributions made in this paper include:

- We propose S^{3}LAM, a novel real-time SLAM system that utilizes 2D Gaussian surfel primitives to achieve both accurate and efficient scene representation.
- We introduce an online adaptive mapping strategy coupled with a novel surfel-based pose estimation algorithm, significantly improving tracking and mapping performance in SLAM under challenging conditions such as severe viewpoint changes.
- We validate the effectiveness of surfel-based tracking through convergence basin analysis. S^{3}LAM demonstrates state-of-the-art performance in both mapping and tracking compared to NeRF-based and 3DGS-based SLAM systems on multiple datasets.

II. Related Work

Dense Visual SLAM aims to achieve simultaneous localization and mapping through reconstructing dense 3D scene representations. Early breakthroughs, e.g., DTAM [17] and KinectFusion [5], laid the foundation for dense SLAM by leveraging photometric consistency and TSDF fusion, respectively, to represent and update the scene. Subsequent work, such as ElasticFusion [18] and DI-Fusion [19], explored a wide range of map representations, including point clouds, surfels, voxel hashing and octrees, to tackle challenges in scalability and accuracy. In parallel, recent advancements have integrated deep learning into traditional frameworks, e.g., SceneCode [20], NodeSLAM [21] and DROID-SLAM [22], to achieve robust camera tracking and mapping. These developments significantly expanded the applications of dense SLAM systems.

NeRF-based SLAM has emerged as a powerful approach for dense visual SLAM, leveraging the advancements in neural radiance fields (NeRF), to jointly optimize scene representation and camera poses. iMAP [7] introduced single-MLP-based scene representation for scalable mapping. NICE-SLAM [23] and Vox-Fusion [8] incorporated hierarchical voxel grids and octree structures, to enhance scalability and precision. Recently, Point-SLAM [24] achieved detailed reconstruction by using neural point clouds with volumetric rendering. However, due to the computational requirements of volume rendering, it faces challenges in real-time performance. Although these methods have demonstrated remarkable accuracy, their reliance on memory-intensive structures and time-consuming rendering limits their scalability and real-time applicability.

3DGS-based SLAM. Recent advancements have integrated 3D Gaussians into RGB-D dense SLAM systems, to represent and render high-quality scenes. The pioneering work [9] demonstrated the effectiveness of 3D Gaussians in photorealistic real-time rendering, but their optimization processes were offline and computationally expensive. To adapt Gaussians for online reconstruction, Yan et al. [10] introduced adaptive expansion and coarse-to-fine tracking. SplaTAM [14] tailored a pipeline to optimize Gaussians with silhouette-guided differentiable rendering. Matsuki et al. [11]

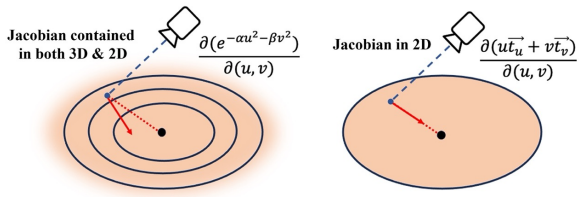


Fig. 3: **Comparison between 3DGS-based and 2D-surfel-based pose optimization. Left**: In 3DGS-based approaches, the optimization relies solely on the Jacobian $\frac{\partial \omega}{\partial T_{CW}}$ to adjust the projected Gaussian function value, which is constrained to move perpendicular to equipotential surfaces. **Right**: The 2D-based Jacobian additionally introduces the radial component $\frac{\partial \hat{\mathbf{t}}_r}{\partial T_{CW}}$ that points directly towards the surfel center.

presented isotropic Gaussian-based tracking, mapping, and rendering for RGBD SLAM. Despite these innovations, real-time performance in large-scale scenes remains challenging. RTG-SLAM [13] sought to unify Gaussian splats for real-time tracking and mapping. Unlike these methods, our approach integrates the Gaussian surfel representation for both mapping and tracking. A concurrent preprint GauS-SLAM [25] also adopted surfels. However, its focus is primarily on local mapping and surfel management, whereas our method emphasizes adaptive surface reconstruction and tracking with strong convergence properties. We begin by briefly reviewing the 2D Gaussian surfel splatting technique, followed by a detailed description of our S^{3}LAM system.

III. Preliminaries

Our S^{3}LAM system adopts 2D Gaussian surfels [15], [26] to represent the scenes, in which surfel splatting is performed by blending Gaussian surfels intersected by rays from image pixels. In the global coordinate system, a surfel is characterized by its center p_k, along with two principal tangential vectors $\mathbf{t}_u$ and $\mathbf{t}_v$, and their corresponding scaling factors s_u and s_v. The normal vector is defined by the cross product $\mathbf{t}_w = \mathbf{t}_u \times \mathbf{t}_v$, and the rotation matrix is represented as $\mathbf{R} = [\mathbf{t}_u, \mathbf{t}_v, \mathbf{t}_w]$. A local point on a Gaussian surfel is parameterized by the (u,v) coordinates:

$$P_k(u,v) = p_k + s_u \mathbf{t}_u u + s_v \mathbf{t}_v v \tag{1}$$

The local coordinates (u,v) are computed by solving the equation $z(x,y,1) = [\mathbf{R}\,|\,\mathbf{t}]P_k(u,v)$, where $z(x,y,1)$ is the point in the camera coordinate system, and $[\mathbf{R}\,|\,\mathbf{t}]$ represents the transformation matrix consisting of the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ in camera extrinsics. Given the local coordinates, the ray-surfel transparency ω_i is calculated as follows:

$$G_k(\mathbf{u}(\mathbf{x})) = \exp(-u^2/2 - v^2/2) \tag{2}$$

$$\omega_i = \alpha_i G_i(\mathbf{u}(\mathbf{x})) \prod_{j=1}^{i-1} (1 - \alpha_j G_j(\mathbf{u}(\mathbf{x}))) \tag{3}$$

Here, $\mathbf{u}(\mathbf{x})$ denotes the local coordinates (u,v) determined by the intersection between the pixel ray from $\mathbf{x}$ and the surfel plane. α_i is the inherent opacity of the i-th surfel. Color, depth, and normal values are rendered using alpha blending, which is computed as the weighted sum of alpha weights ω_k in a front-to-back order of Gaussian surfels:

$$\mathbf{C}(\mathbf{x}) = \sum_{k=1}^{K} \omega_k c_k, \mathbf{D}(\mathbf{x}) = \sum_{k=1}^{K} \omega_k z_k / \sum_{k=1}^{K} \omega_k, \mathbf{N}(\mathbf{x}) = \sum_{k=1}^{K} \omega_k \mathbf{t}_w \tag{4}$$

where c_k represents the color feature vector of the k-th surfel, and z_k represents the z-coordinate of the world point $P_k(u,v)$ transformed into the camera coordinate system. With the Gaussian surfel representation, our map is denoted as $\mathbf{S} = \{(p_k, \mathbf{t}_u, \mathbf{t}_v, s_u, s_v, c_k, \alpha_k)\}$.

IV. Method

We propose S^{3}LAM (short for Surfel Splatting SLAM), a novel RGB-D SLAM system that leverages 2D surfel splatting for high-fidelity and efficient scene mapping. This section is organized as follows: (1) we introduce the 2D Gaussian surfel splatting technique for an efficient map representation; (2) then we present our adaptive reconstruction and surfel management strategy using uncertainty measures, and (3) finally we propose a novel pose optimization algorithm by deriving surfel Jacobians, which improves tracking convergence. The overview of the whole S^{3}LAM system is illustrated in Figure 2.

A. Surfel Splatting Mapping

Given the RGB-D input stream $\{\bar{\mathbf{C}}_t, \bar{\mathbf{D}}_t\}_{t=1}^{N}$, we first compute the local normal map $\bar{N}_t$, by calculating the spatial gradients of the depth map [5]. With the pose T_{CW} estimated by the camera tracking module, the surfels are rendered to determine color $\mathbf{C}_t$, depth $\mathbf{D}_t$ and normal $\mathbf{N}_t$. The mapping optimization loss is defined as a weighted mixture of L_1 losses and cosine similarity:

$$L_{map} = \left\|\mathbf{C}_t - \bar{\mathbf{C}}_t\right\|_1 + \gamma_D \left\|\mathbf{D}_t - \bar{\mathbf{D}}_t\right\|_1 + \gamma_N \left(1 - \mathbf{N}_t \cdot \bar{\mathbf{N}}_t\right) \tag{5}$$

where $\|\cdot\|_1$ denotes L_1 loss over image pixels, $\mathbf{N}_t \cdot \bar{\mathbf{N}}_t$ represents the pixel-wise cosine similarity between the groundtruth normal and estimated normal.

Adaptive Surface Mapping. The original surfel splatting technique [15] is offline, which can effectively reconstruct scene geometry under sufficient optimization iterations, using a large number of surfels for representation and ensuring complete multiview coverage. However, in SLAM systems, the time constraints imposed by real-time mapping and tracking optimization necessitate the use of a reduced number of surfels and a rapidly converging pipeline. Under these constraints, conventional surfel splatting often results in incomplete surface reconstruction, particularly at sharp geometric features such as edges and corners, thereby compromising the accuracy of surface modeling.

To introduce 2D surfel splatting into a real-time SLAM system, we propose an adaptive rendering strategy that can be seamlessly integrated into the surfel rasterization

pipeline, leveraging a depth distortion term $\mathscr{D}_d$ to identify the uncertainty of pixel rays. $\mathscr{D}_d$ is defined as:

$$\mathscr{D}_d = \sum_{i,j} \omega_i \omega_j |z_i - z_j| \tag{6}$$

where ω_i is the same as in Eq. (3), and z_i is the depth of the i-th intersection point. The depth distortion term was originally used to consolidate separate volumes belonging to the same surface [27]. Given a limited number of surfels, regions with high depth distortion typically contain multiple surfels exhibiting large variations in depth z_i and a relatively uniform distribution of blending weights ω_i. In our work, we leverage depth distortion as a separation metric to guide accurate surface reconstruction.

Based on the above observation, we define the uncertainty mask as $\mathscr{D}_d > \tau$, where τ represents the distortion threshold. For pixels with uncertainty exceeding τ, depth and normal values are determined not through opacity-weighted averages, but rather by selecting the surfel with the maximum blending weight. The depth and normal values are then directly determined by the intersection of the ray with the plane defined by this surfel:

$$\mathbf{D}_c(\mathbf{x}) = \arg\max_{\omega_k} z_k, \ \mathbf{N}_c(\mathbf{x}) = \arg\max_{\omega_k} \mathbf{t}_w \tag{7}$$

It is worth noting that the degradation of Gaussian surfels into concrete surfels may inadvertently affect large planar surfaces, potentially resulting in artifacts such as holes on the reconstructed surfaces. To address this issue, we selectively replace the rendered depth $\mathbf{D}$ and normal $\mathbf{N}$ with $\mathbf{D}_c$ and $\mathbf{N}_c$ only when $\mathbf{D}(\mathbf{x}) > \mathbf{D}_c(\mathbf{x})$. This conditional substitution ensures that our adaptive mapping strategy preserves surface integrity while effectively managing uncertainty.

Our adaptive rendering strategy is readily integrated into the surfel rasterization pipeline in both the forward and backward passes, requiring only the retention of dominant surfel information during pixel rasterization. With negligible computational overhead, our strategy simultaneously improves sharp geometry representation and effectively identifies regions of high uncertainty.

Surfel Management. We adopt the surfel addition and deletion operations and their parameters proposed in [10], [13]. During each mapping step, new surfels are added based on three criteria: (1) pixels that remain highly transparent (transmission exceeding a threshold δ_T); (2) pixels exhibiting large depth errors (exceeding a threshold δ_d); and (3) pixels with significant color reconstruction errors (exceeding a threshold δ_c). Erroneous pixels identified by these criteria are uniformly sampled, projected back into 3D space, and incorporated into the surfel set. For surfel deletion, surfels whose accumulated average errors exceed twice the corresponding thresholds are removed in each iteration.

B. Camera Motion Tracking

In the pose tracking module of S^{3}LAM, we utilize the analytical Jacobian of $\mathbf{SE}(3)$ derived from the backward gradient of the surfel splatting. This algorithm is tightly coupled with our model representation, which enhances system robustness and significantly improves convergence speed.

Fig. 4: **Qualitative results of color and depth rendering in S^{3}LAM compared to representative 3DGS-based methods.** Our method exhibits fewer artifacts and clearer borders in the rendered color and depth.

Fig. 5: **Qualitative results of 3D scene reconstruction for the comparison of S^{3}LAM and RTG-SLAM [13].** Our method produces smoother surfaces and finer details in the reconstructed meshes.

Pose Tracking with Surfel Splatting. Surfel splatting has demonstrated its capability for accurate geometric representation. Building on this, we aim to show that pose tracking can also benefit from enhanced surface accuracy. Following [28], we use the Lie algebra framework to derive pose Jacobians, encompassing the 3 degrees of freedom of the rotation group $\mathbf{SO}(3)$ and the 3 degrees of freedom for translation $\mathbf{R}^3$. Pose optimization is formulated as minimizing the following loss with respect to camera pose T_{CW} consisting of rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$:

$$L_{track} = \lambda_C \left\| \mathbf{C}(\mathbf{S}, T_{CW}) - \bar{\mathbf{C}} \right\|_1 + \lambda_D \left\| \mathbf{D}(\mathbf{S}, T_{CW}) - \bar{\mathbf{D}} \right\|_1 . \tag{8}$$

where $\mathbf{C}(\mathbf{S}, T_{CW})$ and $\mathbf{D}(\mathbf{S}, T_{CW})$ are the rendered image and depth map, respectively, derived from the surfel representation $\mathbf{S}$ and the estimated pose $T_{CW} \in \mathbf{SE}(3)$.

For the color loss $L_c = \left\| \mathbf{C}(\mathbf{S}, T_{CW}) - \bar{\mathbf{C}} \right\|_1$, the gradient is propagated through the surfel's central point and its two tangential vectors:

$$\frac{\partial L_c}{\partial T_{CW}} = \sum_{\mathbf{x},k} \frac{\partial L_c}{\partial \mathbf{C}(\mathbf{x})} \frac{\partial \mathbf{C}(\mathbf{x})}{\partial \omega_{k,x}} \frac{\partial \omega_{k,x}}{\partial \hat{\mathbf{q}}_k} \frac{\partial \hat{\mathbf{q}}_k}{\partial T_{CW}} \tag{9}$$

where $\hat{\mathbf{q}}_k = (p_k, \mathbf{t}_u, \mathbf{t}_v)$ denotes the positional parameters of the k-th surfel in camera coordinates. Most terms in our

derivation align with standard 3DGS terminology, and the Jacobians' formulation of the tangent vectors is derived in our demo video.

While the gradient of the color loss with respect to the pose is built upon the same alpha blending framework in both 3D Gaussian spheres and 2D surfels, their formulations for depth rendering differ significantly. In 3DGS, the depth is computed as a weighted average of the projected Gaussian centers: $\mathbf{D}(\mathbf{x}) = \sum_i \omega_i \hat{\mathbf{p}}_{i,z}$ where $\hat{\mathbf{p}}_{i,z}$ is the z-coordinate of the i-th Gaussian center after being transformed under the current viewpoint [10], [11]. Due to the unoriented nature of Gaussian spheres, the camera pose rotation is governed solely by the Jacobians of the alpha weights, $\frac{\partial \omega}{\partial T_{CW}}$. These Jacobians exhibit directional bias due to the anisotropic characteristics of Gaussian spheres, which inherently constrains the camera to move perpendicular to the equipotential surfaces of the Gaussian function, as illustrated in Figure 3 left.

Unlike 3DGS, depth in 2D surfel splatting [15] is computed as the blending of ray-surfel intersection points: $\mathbf{D}(\mathbf{x}) = \sum_i \omega_i \hat{P}(\mathbf{u}(\mathbf{x}))_{i,z}$, for which the gradient is additionally influenced by the intersection-center residual:

$$\begin{aligned}\frac{\partial \mathbf{D}(\mathbf{x})}{\partial T_{CW}} &= \frac{\partial \mathbf{D}(\mathbf{x})}{\partial \omega_{i,x}} \frac{\partial \omega_{i,x}}{\partial T_{CW}} + \frac{\partial \mathbf{D}(\mathbf{x})}{\partial \hat{P}(u(x))} \frac{\partial \hat{P}(u(x))}{\partial T_{CW}} \\ &= \frac{\partial \mathbf{D}(\mathbf{x})}{\partial \omega_{i,x}} \frac{\partial \omega_{i,x}}{\partial T_{CW}} + \frac{\partial \mathbf{D}(\mathbf{x})}{\partial \hat{P}(u(x))} \left(\frac{\partial \hat{p}_i}{\partial T_{CW}} + \underbrace{\frac{\partial \hat{\mathbf{t}}_r}{\partial T_{CW}}}_{\text{radial grad}} \right)\end{aligned} \tag{10}$$

where $\hat{P}(u(x))$ denotes the transformed intersection point between pixel ray and surfel plane, and $\hat{\mathbf{t}}_r = R(us_u\mathbf{t}_u + vs_v\mathbf{t}_v)$ represents the vector from surfel center to ray-surfel intersection point, satisfying the relation $\hat{P}(u(x)) = \hat{p}_i + \hat{\mathbf{t}}_r$. Compared to the alpha-blending-style depth rendering in 3DGS, the Jacobian of surfel-based depth rendering includes an additional radial vector component, $\hat{\mathbf{t}}_r$. This component is critical for locally optimizing the pixel ray, enabling it to move closer to or farther from the surfel center by controlling the camera's rotation, as illustrated in Figure 3 right.

From a statistical perspective, the radial gradient provided by oriented surfels acts as a guidance mechanism for aligning the reconstructed surface with the ground-truth target view. This gradient directly influences the camera's orientation, encouraging the pose to rotate either perpendicular or parallel to the underlying plane. As a result, our surfel-based pose optimization exhibits remarkable tracking robustness even with very few frame-to-frame overlap, demonstrating a wide convergence basin and enabling stable motion tracking under significant viewpoint variations.

Using the derived pose Jacobian, we iteratively optimize the camera pose as new frames arrive, taking advantage of the geometrically accurate representation provided by the oriented surfels. Following RTG-SLAM [13], we integrate the Iterative Closest Point (ICP) method [5] in S^{3}LAM as an optional camera tracking module for synthetic datasets, which utilizes high-fidelity depth information.

TABLE I: **Comparison of geometry accuracy on Replica.** The P and R represent precision and recall respectively. The best metric is highlighted in **bold**, and the second-best metric is underlined.

Method	Acc.↓	P.↑	Comp.↓	R.↑	F1↑	L1↓
NICE-SLAM [23]	2.84	84.4	2.31	85.0	84.7	2.64
Point-SLAM [24]	**0.61**	**99.9**	2.42	86.9	**92.7**	**0.44**
MonoGS [11]	2.72	75.6	3.29	76.6	75.6	3.56
SplaTAM [14]	2.88	73.9	3.57	71.7	72.8	0.70
RTG-SLAM[13]	0.75	98.9	2.81	82.8	90.0	1.71
Ours	1.06	95.2	**2.19**	**88.6**	91.9	0.47

Keyframing and Optimization Strategies. Our keyframing strategy is based on the camera motion [29]. The initial frame is added to the keyframe list, and subsequent frames are designated as keyframes, if their relative rotation exceeds a threshold δ_r or their relative translation surpasses δ_t. These newly identified keyframes are appended to the keyframe list.

The surfel map is optimized every M frames using the most recent frames. Additionally, when a new keyframe is added, the surfel map is optimized using the former keyframes. After completing the entire SLAM procedure, the surfel splatting map is further refined by optimizing with all recorded keyframes, employing ten times the number of keyframe iterations to achieve the final accuracy.

V. Experiments

A. Experimenal Settings

Datasets. We conducted experiments on three representative datasets: the synthetic Replica [30], the real-world TUM-RGBD [31] and ScanNet++ [32]. 3D reconstruction experiments are performed on the Replica dataset due to its availability of ground-truth reconstructed meshes. Pose estimation experiments are conducted on all three datasets. For ScanNet++, we selected four sequences: 8b5caf3398 (S0), b20a261fdf (S1), and f34d532901 (S2), as S0 and truncated S1 have been previously benchmarked in SLAM research [14].

Implementation Details. To implement our proposed adaptive surface mapping and pose Jacobian algorithms, we revised the 2D Gaussian splatting CUDA code in [15]. For the coupled pose estimation algorithm, map optimization was performed every 6 frames for 50 iterations, and pose optimization involved 50 iterations on Replica. We set the mapping hyper-parameters γ_D to 1.0, γ_N to 0.1, τ to $5E-6$, δ_T to 0.5, δ_d and δ_c to 0.1. All experiments were performed on a Platinum 8375C CPU with an RTX3090 GPU.

Baselines. We compare the performation of S^{3}LAM with (1) recent NeRF-based RGB-D SLAM methods: NICE-SLAM [23] and Point-SLAM [24], and (2) state-of-the-art 3D Gaussian splatting-based methods: Gaussian Splatting SLAM (MonoGS) [11], SplaTAM [14], and RTG-SLAM [13]. The official implementations of these baselines were reproduced in the same environment.

TABLE II: **Comparison on Run-time Performance.** Metrics are obtained from run-time evaluations on the Office0 sequence of the Replica dataset.

	FPS ↑	Memory ↓	Model Size ↓
Point-SLAM[24]	0.32	11189 MB	2864 MB
SplaTAM[14]	0.34	25716 MB	310 MB
MonoGS[11]	0.87	10340 MB	17.4 MB
RTG-SLAM[13]	8.95	3444 MB	46.7 MB
Ours	8.12	4200 MB	43.2 MB

B. Results

3D Scene Reconstruction. Following NICE-SLAM [23], we evaluate different methods using the following metrics: accuracy (cm), precision (the proportion of reconstructed points with an accuracy below 3cm, %), completion (cm), recall (the proportion of ground-truth points with a completion below 3cm, %), F1-score, and L1 depth error (cm). Consistent with NeRF-based SLAM methods, we generate the scene mesh using the TSDF-Fusion algorithm [33] after processing the RGB-D input sequence for S^{3}LAM and MonoGS. For RTG-SLAM and SplaTAM, we adopt their densification method to create the scene's point cloud as in [13].

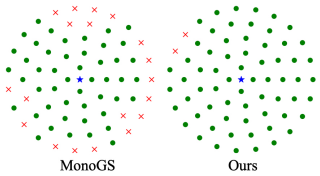
The experimental results are summarized in Table I. Our method achieves state-of-the-art performance in 3D scene reconstruction by balancing precision and recall. This balance ensures that the reconstructed meshes are not only densely populated but also closely aligned with ground-truth vertices. By combining Gaussian surfel representations with our novel adaptive mapping strategy, S^{3}LAM achieves highly accurate depth estimation from rendered maps, significantly enhancing the overall SLAM performance. Quantitative evaluations in Table I demonstrate that S^{3}LAM outperforms 3DGS-based SLAM systems while achieving reconstruction quality comparable to that of leading NeRF-based SLAMs. Notably, as shown in Table II[1], S^{3}LAM has substantial advantages in both computational efficiency and memory consumption compared to the NeRF-based SLAM.

We also conduct a qualitative comparison on image rendering, depth rendering, and mesh reconstruction, as shown in Figures 4 and 5. In both rendered image and depth maps, S^{3}LAM generates significantly fewer artifacts compared to representative 3DGS-based SLAMs. In mesh reconstruction, the results of S^{3}LAM produces more detailed object boundaries and smoother scene surfaces, underscoring the geometric accuracy of S^{3}LAM.

Pose Tracking. We evaluate baseline methods and our two pose tracking strategies on the Replica dataset. The experimental results are summarized in Table III. In S^{3}LAM, we refer to the method *pose tracking with surfel splatting* and the ICP-based tracking as **Ours-Coupled** and **Ours-ICP**, respectively. On the Replica dataset, due to high-quality depth data and small pose variations, the ICP method achieves very good performance. Compared to other coupled methods using Gaussian gradients, our surfel-based tracking method demonstrates superior performance.

[1]Since our experiments are conducted on a less powerful GPU (RTX 3090) compared to the RTX 4090 used in the original paper, the reported FPS values may not be directly comparable.

Fig. 6: **Qualitative ablation results:** Depth reconstruction without the adaptive strategy using mean depth (**Left**), with our proposed distortion-based adaptive strategy (**middle**), and the ground-truth depth (**right**).



	Seq1	Seq2	Seq3
MonoGS	0.72	0.46	0.55
Ours	**0.97**	**0.77**	**1.00**

Fig. 7: **Comparison on convergence basin (left) and tracking converging rate (right).**

For the TUM-RGBD and ScanNet++ datasets, where depth images are noisy and inter-frame motions can be large, we use only the coupled surfel-based tracking without ICP, as it offers better convergence and smaller drift. The results on the TUM-RGBD dataset are shown in Table IV. It is worth noting that the significant noise and incompleteness of depth data in TUM-RGBD have small impact on MonoGS [11], which primarily relies on color information. On the other hand, our method outperforms other NeRF-based and 3DGS-based RGB-D SLAM methods, and achieves competitive performance with MonoGS [11].

The tracking results on the ScanNet++ dataset are shown in Table V. ScanNet++ features abrupt teleportations and rotations in its trajectories, causing many methods to either fail to track camera motion or track only partial camera motion. SplaTAM [14] was originally tested on S0 and truncated S1 from the 0-th frame to the 360-th frame, due to intense relative movement in the subsequent frames.

In contrast, our method demonstrates strong tracking convergence even when adjacent frames exhibit very small overlap, achieving state-of-the-art performance. For example, in S1 between frames 363 and 364, where a significant rotation causes other methods to fail, our method successfully tracks the large viewpoint change. Thanks to our proposed surfel-based pose optimization, S^{3}LAM successfully converges to the correct pose, which is achieved through the Jacobian formulation of the oriented surfels, as derived in our pose tracking method. In the next section, a detailed analysis of the tracking convergence is provided.

C. Convergence Basin Analysis

We follow [34], [11] to evaluate pose convergence. To simulate SLAM's limited perception, we train the 2DGS scene using only a restricted 3×3 grid of views with 0.1m spacing. We sample initial camera positions from 0.2m to

TABLE III: **Comparison of tracking performance on Replica with ATE (unit: cm).** * denotes SLAMs utilizing ICP algorithm for tracking.

Method	Rm 0	Rm 1	Rm 2	Off 0	Off 1	Off 2	Off 3	Off 4	Avg.
NICE-SLAM[23]	0.97	1.31	1.07	0.88	1.00	1.10	1.10	1.13	1.06
Point-SLAM[24]	0.54	0.43	0.34	0.36	0.45	0.44	0.63	0.72	0.49
SplaTAM[14]	0.47	0.42	0.32	0.46	0.24	0.28	0.39	0.56	0.39
MonoGS[11]	0.64	0.34	0.37	0.47	0.66	0.25	**0.16**	2.36	0.66
RTG-SLAM[13]*	0.20	**0.18**	0.13	0.22	**0.12**	**0.22**	0.20	**0.19**	**0.18**
Ours-ICP*	**0.19**	**0.18**	**0.11**	**0.18**	0.13	0.29	0.17	0.22	0.21
Ours-Coupled	0.29	0.68	0.39	0.22	0.15	0.48	0.46	0.35	0.38

TABLE IV: **Comparison of tracking performance on TUM-RGBD with ATE (unit: cm).**

Methods	fr1/desk	fr2/xyz	fr3/office	Avg.
NICE-SLAM [23]	4.26	6.19	6.87	5.77
Point-SLAM [24]	4.34	1.31	3.48	3.04
SplaTAM [14]	3.35	1.24	5.16	3.25
MonoGS [11]	**1.52**	1.58	**1.65**	**1.58**
Ours	2.38	**1.16**	2.26	1.93

TABLE V: **Comparison of tracking performance on ScanNet++ with ATE (unit: cm).** The S1* denotes the truncated sequence for S1.

Methods	S0	S1*	S1	S2
Point-SLAM[24]	411.9	1000.8	829.4	1888.4
MonoGS[11]	141.5	106.9	122.7	147.7
SplaTAM[14]	0.63	1.90	70.93	2.05
Ours	**0.35**	**0.42**	**0.51**	**1.11**

1.6m from the target view (center view of training data) and run 1,000 pose optimization steps using the RGB and depth of the target view. We count the run as successful if the final pose is within 1cm of the target.

We compare our method with a representative 3DGS-based pose optimization [11] on three Replica sequences. Figure 7 shows quantitative success rates[2] and the visualized convergence basins. Our method achieves better convergence even when initial poses are far from the target, with very small view overlap. This improved behavior explains the strong tracking performance on ScanNet++, indicating that pose tracking with oriented surfel primitives significantly expands the convergence basin under large viewpoint changes.

D. Ablation Study

We conducted an ablation study to evaluate the impact of distortion-adaptive rendering. The quantitative results are presented in Table VI, comparing our method with 3D Gaussian splatting, 2D Gaussian splatting reconstructions based on mean depth and median depth strategies [15]. Mapping with 3D Gaussian primitives yields suboptimal depth estimation and reconstruction performance. Similarly, without using depth distortion as a measure of mapping uncertainty, surfel splatting mapping also results in suboptimal reconstruction.

[2]Since MonoGS [11] does not specify exact sequences, our selected sequences may yield different results.

TABLE VI: **Ablation study for adaptive mapping.**

	Acc.↓	P.↑	Comp.↓	R.↑	F1↑	L1↓
3DGS	2.80	3.71	74.8	70.1	72.3	2.21
Mean	1.41	91.5	**2.09**	**88.9**	90.1	0.53
Median	1.10	94.7	2.35	87.1	90.6	0.71
Adaptive	**1.06**	**95.2**	2.19	88.6	**91.9**	**0.47**

TABLE VII: **Ablation study comparing tracking accuracy with ATE (unit: cm) on ScanNet++.**

	S0	S1*	S1	S2
w/o depth loss	162.5	138.5	168.4	150.1
w/o radial	224.0	331.4	567.2	200.8
Ours	**0.35**	**0.42**	**0.51**	**1.11**

Interestingly, reconstruction based on mean depth achieves higher recall, while median depth provides better precision. Our method, which combines average depth blending with dominant surfel-determined depth rendering, successfully captures the benefits of both strategies, achieving comparable precision and superior recall. Qualitative results shown in Figure 6 further highlight the degradation in reconstruction quality when distortion awareness is omitted, underscoring the importance of incorporating depth distortion into the mapping process.

For the ablation of pose optimization, we evaluate performance without the depth tracking loss and with the radial Jacobian explicitly removed from the tracking on ScanNet++. The results in Table VII validate that the whole pose tracking design leads to better convergence and accuracy.

E. Real-world Experiments

Beyond evaluations on public SLAM datasets, we further validated our method on real-world tracking and mapping tasks using self-captured data under both vehicle-mounted and hand-held scenarios, as illustrated in Fig. 8. Data were collected with the Orbbec Astra Plus camera across diverse environments. Representative samples are shown in Fig. 8, and comprehensive experimental results are provided in our demo video.

VI. Conclusion

In this paper, we present S^{3}LAM, a novel SLAM system that employs 2D surfel splatting as the core representation. Our approach achieves high-fidelity surface reconstruction

(a) Our Devices (b) Platform (c) Captured Image

Fig. 8: The real-world platform consists of a RGB camera, a robot platform, the experimental environment.

and high-convergence pose tracking, even in challenging scenarios. A current limitation is its reduced performance with low-quality sensor inputs and limited generalizability to large-scale environments. In future work, we aim to enhance the system's robustness under noisy sensing conditions and expand its applicability to broader scenes.

REFERENCES

[1] F.-x. Chen, Y. Tang, C. Tai, X.-p. Liu, X. Wu, T. Zhang, and L. Zeng, "Fusednet: End-to-end mobile robot relocalization in dynamic large-scale scene," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4099–4105, 2024.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[3] F. Ruetz, E. Hernández, M. Pfeiffer, H. Oleynikova, M. Cox, T. Lowe, and P. Borges, "Ovpc mesh: 3d free-space representation for local ground vehicle navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8648–8654.

[4] A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration," *ACM Transactions on Graphics 2017 (TOG)*, 2017.

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

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

[8] X. Yang, H. Li, H. Zhai, Y. Ming, Y. Liu, and G. Zhang, "Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 499–507.

[9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 139–1, July 2023.

[10] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li, "Gs-slam: Dense visual slam with 3d gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2024, pp. 19 595–19 604.

[11] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 039–18 048.

[12] J. Hu, X. Chen, B. Feng, G. Li, L. Yang, H. Bao, G. Zhang, and Z. Cui, "Cg-slam: Efficient dense rgb-d slam in a consistent uncertainty-aware 3d gaussian field," in *European Conference on Computer Vision*, Springer. Springer, 2025, pp. 93–112.

[13] Z. Peng, T. Shao, Y. Liu, J. Zhou, Y. Yang, J. Wang, and K. Zhou, "Rtg-slam: Real-time 3d reconstruction at scale using gaussian splatting," in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[14] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. 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*. IEEE, 2024, pp. 21 357–21 366.

[15] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2d gaussian splatting for geometrically accurate radiance fields," in *ACM SIGGRAPH 2024 conference papers*. ACM, 2024, pp. 1–11.

[16] P. Dai, J. Xu, W. Xie, X. Liu, H. Wang, and W. Xu, "High-quality surface reconstruction using gaussian surfels," in *ACM SIGGRAPH 2024 Conference Papers*. ACM, 2024, pp. 1–11.

[17] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*, IEEE. IEEE, 2011, pp. 2320–2327.

[18] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[19] J. Huang, S.-S. Huang, H. Song, and S.-M. Hu, "Di-fusion: Online implicit 3d reconstruction with deep priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 2021, pp. 8932–8941.

[20] S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison, "Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 11 776–11 785.

[21] E. Sucar, K. Wada, and A. Davison, "Nodeslam: Neural object descriptors for multi-view shape reconstruction," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 949–958.

[22] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[23] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "Nice-slam: Neural implicit scalable encoding for slam," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE, 2022, pp. 12 786–12 796.

[24] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2023, pp. 18 433–18 444.

[25] Y. Su, L. Chen, K. Zhang, Z. Zhao, C. Hou, and Z. Yu, "Gaus-slam: Dense rgb-d slam with gaussian surfels," 2025. [Online]. Available: https://arxiv.org/abs/2505.01934

[26] W. Zhang, H. Xiang, Z. Liao, X. Lai, X. Li, and L. Zeng, "2dgs-room: Seed-guided 2d gaussian splatting with geometric constrains for high-fidelity indoor scene reconstruction," *arXiv preprint arXiv:2412.03428*, 2024.

[27] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022, pp. 5470–5479.

[28] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," 2021. [Online]. Available: https://arxiv.org/abs/1812.01537

[29] Y.-P. Cao, L. Kobbelt, and S.-M. Hu, "Real-time high-accuracy three-dimensional reconstruction with consumer rgb-d cameras," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 5, pp. 1–16, 2018.

[30] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, "The replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[31] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, IEEE. IEEE, 2012, pp. 573–580.

[32] C. Yeshwanth, Y.-C. Liu, M. Nießner, and A. Dai, "Scannet++: A high-fidelity dataset of 3d indoor scenes," in *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 2023, pp. 12–22.

[33] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[34] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 22–31.